Structured Approach to Industrial Control System Design

Birga Syska

A Thesis submitted in partial fulfilment of the requirements of the

University of Glamorgan / Prifysgol Morgannwg

for the degree of

Doctor of Philosophy

School of Technology

July 2004

Approval for Submission of a Thesis

Student:.................Birga Syska...

Award:.................Doctor of Philosophy..

This is to confirm that a thesis entitled:

Structured Approach to Industrial Control System Design

has been approved for submission by the student's supervisory team and can therefore be forwarded to the examiners.

Signed ....

Director of Studies

Date ....

Confirm Thesis

UNIVERSITY OF GLAMORGAN
PRIFYSGOL MORGANNWG

(This form should be typewritten)

Candidate's Declaration Form

Note: This form must be submitted to the University with the candidate's thesis (10.5 of the Regulations refers)

Name of Candidate: *Birga Syska*

Degree for which thesis is submitted: *PhD*

1. **Statement of advanced studies undertaken in connection with the programme of research** (Regulation 4.1 refers)

N/A

2. **Concurrent registration for two or more academic awards** (Regulation 4.8 refers)

either * I declare that while registered as a candidate for the University of Glamorgan's research degree, I have not been a registered candidate or enrolled student for another award of the University or other academic or professional institution.

~~or * I declare while registered for the University of Glamorgan's research degree, I was, with the University's specific permission, a *registered candidate/*enrolled student for the following award.~~

3. **Material submitted for another award**

either I declare that no material contained in the thesis has been used in any other submission for any academic award.

~~or I declare that the following material contained in the thesis formed part of a submission for the award of~~

~~*(state award and awarding body and list the material below)*~~

Signature of candidate: .... Date: 29/09/04

*** delete as appropriate***

Dedication

To my parents and my daughter Laura

Acknowledgement

First of all I would like to thank Prof. John Ward, the Director of Studies of this research project, for his support and help in my research work.

I am most grateful to my supervisor team Prof. Dr-Ing. Reimar Schumann and Dr. Guiliano C. Premier for their enthusiastic supervision, help, patience and encouragement during the course of my PhD programme.

Many thanks to my research colleague Dr. Steffen Körner who provided the basics for my research work and for all the stimulating discussions during our time together at the Fachhochschule Hannover. I would also like to thank Peter Hoffmann for his support in the control laboratory and Dirk Kruse for programming the OPC server/client application.

Grateful thanks as well to my friend Dr. Ales Belic for his support in solving my MATLAB™ problems and his uplifting notes. I would also like to thank my friends Michael and his family and Birgit and her family for their support in this phase of my life.

Furthermore, I would like to thank my employer Körting Hannover AG for allowing me the time to write up this thesis.

My thanks also to Mr. Keith Baker as the previous Director of Studies for his initial support in this research project.

Finally, I would like to thank my family, especially my parents, for their support, encouragement and care of my daughter Laura. Last, but most certainly not least, a very special thank-you to my daughter Laura, for her patience with me and her understanding of my work and absences.

Abstract

The design of complex control systems for industrial processes is in general still based on experience and trial and error rather than on systematic process analysis and control system design. Typically control systems in the process industry comprise either single PID control loops with clear association of the process input and controlled signal or standard control schemes developed by intuition through the years (e.g. cascade control using an underlying flow control loop). Although many of these control schemes seem to work rather satisfactorily, in most major control systems several poorly tuned or switched off controllers are encountered leading to unsatisfactory process behaviour or manual operation of the process.

The subject of this work is the development of a *structured approach to industrial control system design* which makes the potential of systematic process analysis and controller design methods developed by control theory available for industrial users with limited control experience. This aspect has been addressed within the collaboration research project between the University of Glamorgan and the Fachhochschule Hannover, of which the work presented in this thesis is a substantial part.

Therefore, the ICACSD (Industrial Computer Aided Control System Design) scheme has been developed to allow the design of PID based control structures for nonlinear single- and multivariable processes that are as simple as possible and as good as required. Beyond this an industrial standardised controller design procedure for nonlinear and multivariable processes has been elaborated. For the validation of the proposed approach, a prototype control system design tool has been programmed, which can be integrated into the ICACSD scheme using a block-oriented simulation environment.

The approach for industrial control system design presented shows the benefits of applying advanced control system design methods which are usable by industrial users when provided with an intuitive and usable graphical user interface. In order to validate the work the proposed control design procedure has been made accessible in the form of a software prototype with an ergonomically designed graphical user interface allowing easy application of the developed methods. The

prototype realisation for the *Industrial Computer Aided Control (ICAC)* toolbox supplies the new structured approach to control system design for industrial processes within a block-oriented simulation environment.

Table of Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

ANN	Artificial Neural Network
APCDE	Advanced Process Control Development Environment
CACE	Computer Aided Control Engineering
CACSD	Computer Aided Control System Design
CAE	Computer Aided Engineering
CD block	Control Design block
CSGPC	Constrained Stable Generalised Predictive Control
D	Derivative part of controller
DMC	Dynamic Matrix Control
EPSAC	Extended Prediction Self-Adaptive Control
GMA	Gesellschaft für Mess- und Automatisierungstechnik (German Measurement and Automation Society)
GPC	Generalised Predictive Control
GUI	Graphical User Interface
I	Integral part of controller
IAE	Integral Absolute Error
ICAC	Industrial Computer Aided Controller design
ICACSD	Industrial Computer Aided Control System Design
ICAI	Industrial Computer Aided Identification
ID block	Identification block
IFAC	International Federation of Automation Control
ISE	Integral Squared Error
ITAE	Integral Time Absolute Error

ITSE	Integral Time Squared Error
LKR	Laboratory air-conditioning plant (Labor Klima Regelung)
LMPC	Linear Model Predictive Control
MAC	Model Algorithm Control
MES	Model Evolution Scheme
MGWM	Modified Generalised Weighted Mean
MIMO	Multiple Input, Multiple Output
MPC	Model Predictive Control
MPHC	Model Predictive Heuristic Control
NGPC	Nonlinear Generalised Predictive Control
OLE	Object Linking and Embedding
OPC	OLE for Process Control
P	Proportional part of controller
PD	Proportional plus Derivative
PI	Proportional plus Integral
PCS	Process Control System
PID	Proportional plus Integral plus Derivative
QDMC	Quadratic Dynamic Matrix Control
SC	Static Characteristic
SISO	Single Input, Single Output
SNIP	Standardised Nonlinear Identification Procedure

Latin Symbols

$\underline{\mathbf{A}}$	system matrix
$\underline{\mathbf{b}}$	input vector
$\underline{\mathbf{B}}$	input matrix
$\underline{\mathbf{c}}^T$	transpose of vector **c** (output vector)
$\underline{\mathbf{C}}$	output matrix
$\underline{\mathbf{D}}$	input-output matrix
e	control deviation
e_c	control deviation of PID control loop
e_{ss}	control deviation of the state space control loop
$G_c(s)$	transfer function of controller
G_{ii}	main process element
G_{ij}	coupling process elements
$G_p(s)$	transfer function of process
$G_{pd}(s)$	transfer function of PD controller
$G_{pi}(s)$	transfer function of PI controller
K_d	derivative coefficient
K_i	integral coefficient
K_p	proportional gain
K_u	ultimate gain
$\mathbf{Q_B}$	observable matrix
R_{ii}	main controller
R_{ij}	decoupling controller
$\underline{\mathbf{T}}$	transformation matrix
T_d	derivative action time (or 'rate time') = K_d/K_p

T_i	integral action time (or 'reset time') = K_p/K_i
T_n	filter's time constant
u(s)	input signal (control variable)
u	input vector
u_c	process input of the PID control loop
u_F	airflow voltage at the LKR
u_{ss}	process input of the state space control loop
u_φ	voltage on the nebulizer at the LKR
u_ϑ	heating voltage at the LKR
w	set point
x	state vector
y(s)	output signal (controlled variable)
y	output vector

Greek Symbols

φ_1	humidity after air inlet at the LKR
φ_2	humidity after heating coil at the LKR
φ_3	humidity in the mixing chamber of the LKR
ϑ_1	temperature after air inlet at the LKR
ϑ_2	temperature after heating coil at the LKR
ϑ_3	temperature in the mixing chamber of the LKR

1 Introduction

World-wide competition in process industries requires processes which are made more profitable by improving quality, increasing throughput, decreasing maintenance effort and other operating costs while maximising profits. Moreover environmental problems demand the best possible use of resources and the minimisation of waste. Conventional standard PID controllers, which are widely used in process industries, cannot meet these demands in all cases because of increasingly complex process designs that frequently exhibit nonlinear behaviour and couplings (Hahn and Nöth, 1993). Nevertheless, it has been shown in some practical applications that modern process control strategies are capable of tackling even very complex and difficult control tasks quite well (see for example, Grawthrop and Ponton, 1996). Some widely used process control systems (PCS) for example, provide the necessary functionality but their potential for improvement of control strategies is rarely exploited because these require additional time and methods and are not suitable for industrial users without control system design expertise.

With increasing demands on process efficiency, product quality and environmental compatibility the need for better control and process optimisation leads to the question of how the potential of systematic process analysis and controller design methods developed by control theorists over recent decades can be made available to the industrial control engineer to improve the situation just described. One of the possible answers is the use of industrial computer aided control system design (CACSD) tools tailored to the control design tasks and knowledge level of industrial engineers. The aim is to hide the complexity of theoretical methods under an industrial user interface and to adapt the design results to the means of realisation in industrial process control systems.

1.1 The Need for a New Approach to Industrial Controller Design

For economical and ecological optimisation, i.e. process efficiency, product quality and environmental compatibility, process industry is urged to invest in automation strategies which utilise the whole potential of control. The subject of this work is the development of a new structured approach to control design techniques, which enable process engineers and process personnel to easily design control systems even for multivariable nonlinear processes utilising their knowledge of the process under investigation.

Particularly where control experts are rare and unavailable – this being the case in many small and medium size companies – this approach can provide a solid base for automated controller design. To satisfy these needs it is essential to support this approach using computer aids, aimed at simplifying the application.

Numerous control system design programs have been developed as part of Computer Aided Control System Design (CACSD) tools (Schumann, 1998; Schmid, 1993). Generally these programs are excellent test beds, providing a comprehensive collection of sophisticated aids for the main CACSD tasks such as process modelling, controller design and in some

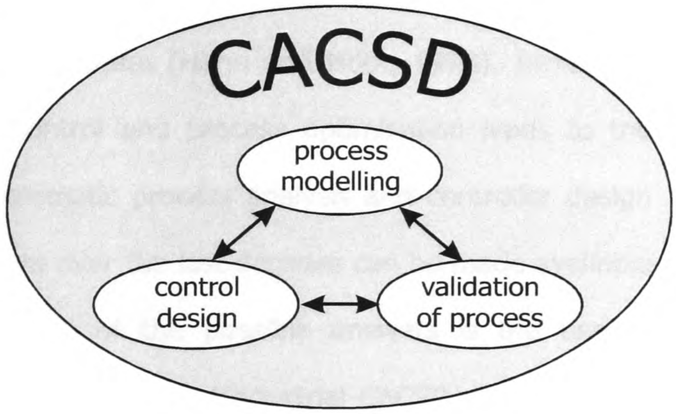


Figure 1-1 CACSD tasks

cases also the validation of the process (Figure 1-1). Nevertheless, these programs do not address practical design aspects sufficiently, because these programs were mainly developed in and for an academic environment.

Process engineers explained in CACSD workshops of the GMA - VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (VDI/VDE Society for Measurement and Automatic Control) GMA-Aussprachetage Rechnergestützter Entwurf von Regelungssystemen 1997-Kassel, 2001-Dresden (GMA-Bericht 32, 1997; GMA-Bericht 36, 2001) that the design of complex control systems for industrial processes is in general still based on experience and trial and error rather than on systematic process analysis and control system design. Typically control systems in the process industry comprise either single PID control loops with clear association of a process input with a control signal (e.g. heating and temperature) or standard control schemes developed by intuition through the years (e.g. cascade control using an underlying flow control loop). Although many of these control schemes seem to work satisfactorily, in most major control systems several poorly tuned or "switched off" controllers are encountered. This leads to unsatisfactory process behaviour or manual operation of the process (Hahn and Nöth, 1993). Increasing demands and the need for better control and process optimisation leads to the question of how the potential of systematic process analysis and controller design methods developed by control theorists over the last decades can be made available to the industrial control engineer. One of the possible answers is the use of Industrial Computer Aided Control System Design (Industrial CACSD) tools tailored to the control design tasks and knowledge levels of industrial engineers. The aim is to hide the complexity of the theoretical methods under an industrial user interface and to adapt the design results to the means of realisation widely used in industrial process control systems.

1.2 Contributions of this Thesis

The aim of this research project is to develop a streamlined approach that concentrates on the determination of improved PID-based control schemes by advanced controller design methods, especially in the case of nonlinear multivariable systems where the variety of possible control structures is virtually endless. Nevertheless, it is desirable that non-expert users should have the ability to cope with nonlinear multivariable processes. Therefore, in this work, a specific class of control structures has been selected to support a standardised controller design procedure for a variety of nonlinear multivariable processes. The new approach centres on solutions for the 'area'-engineer in the process industry and not on the control specialist. It leads to an appropriate class of controllers including nonlinear multiple-input multiple-output (MIMO) controllers. To validate the new approach a software prototype has been developed for *Industrial Computer Aided Control (ICAC)*.

1.2.1 Relevance to Industrial Applications

For the development of a practically applicable yet progressive approach it is especially important to understand the specific constraints imposed for the application of computer aided control system design in industry. Interviews with process engineers during CACSD workshops and conferences of the GMA - VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (VDI/VDE Society for Measurement and Automatic Control) revealed that most process engineers and process personnel in the process industry are often extremely short of time and relatively inexperienced at modelling, simulation and control. At first glance this contrasts with the observation that process engineers or process personnel in industry have a fairly good knowledge of their processes. They are well equipped to know what to do if the process behaviour becomes unstable or insufficiently controlled. Mostly, however, this knowledge is heuristic knowledge from experience and training of the process engineers or process personnel and cannot be used systematically or define it in a mathematical way which would be suitable for traditional control system design. Therefore, the newly proposed structured approach to control design considers the different aspects necessary to allow successful industrial application. In this sense the following properties were specifically relevant for the *ICAC* prototype development:

- The software tool has been designed to be straightforward in use without requiring special training or repeated familiarisation phases. A guided tour to controller design is provided that is geared to the industrial user's knowledge. This implies that the software realisation of this controller design approach has been created in such a way that the available functionality is task-oriented and intuitive to use thus making every step in the controller design procedure as simple as possible.
- As part of an integrated industrial interface concept different user levels of complexity and functionality are provided. For example a standardised controller design procedure is provided for process personnel, that displays the results in the time domain according to the user's understanding. However, control specialists need more degrees of freedom and functions while performing the controller design task in order to utilise their whole creative potential.

- The software tool utilises selected controller design techniques, which have been tailored to the needs of industrial 'area'-engineers with broad but shallow control knowledge in an effective and economic way. This means that the complexity of advanced methods is hidden behind an easily accessible user interface by utilising sensible defaults.

1.2.2 Relevance to Controller Design Theory

In order to achieve the research objectives as outlined before, it was necessary to access a wide range of different but complementary research fields. Consequently this research has a broad view and utilises available schemes and methods where possible.

However, with the focus set on industrial application a new Industrial CACSD scheme has been developed that formalises a widely used practical approach to multivariable control system design as it is carried out in process industry. Additionally it introduces an Industrial CACSD procedure that is aimed at a specific class of controller. This procedure has set the frame for the elaboration of a standardised control system design procedure that is aimed at easy application, separating the design of the static and the linear dynamic parts of the control system.

1.2.3 Relevance to Future Developments

This research project is a part of the collaborative development of an Industrial CACSD tool carried out at the University of Glamorgan and the Fachhochschule Hannover. Hence the software concept has been created to be open, expandable and modular to allow the collaborating partners to incorporate new ideas in the future.

The resulting process identification modules of this developing Industrial CACSD tool are contained in *ICAI (Industrial Computer Aided Identification)* (Körner, 1999) and Model^{ing} (Strickrodt, 1997). *ICAC* is able to incorporate the results from the *ICAI* software tool directly. The combined modules *ICAI* and *ICAC* have the potential for improved commissioning in process industry. These modules have been connected to a commercial process control system to enable prototype tests in the industrial settings. Due to a close co-operation with ABB Automation, Freelance 2000 ® is the process control system (PCS) used. The connection has been realised with an OPC server/client application.

1.3 Organisation of this Thesis

The organisation of this thesis follows a top down approach starting with a broad description of the background that initiated the work and going into details, where necessary in order to present the work carried out to accomplish a prototype of the industrially suitable controller design software.

The thesis is organised as follows:

- The *second chapter* reviews current control design in process industry and technological development in the field of CACSD. It also demonstrates the urgent need for a control system design tool aimed at industrial users as part of a CACSD tool.

- In the *third chapter* methods and methodologies of available software tools for complex control system design are critically reviewed with respect to industrial applicability
- Based on the foregoing work a concept for an Industrial CACSD scheme is presented in the *fourth chapter*.
- A procedure for the preparation of nonlinear and multivariable process models is described in the *fifth chapter*
- In the *sixth chapter* a proposal for an industrial control system design procedure is introduced
- Based on the results gained so far, the *seventh chapter* describes the ICAC prototype realised in MATLAB™.
- This prototype has also been used to validate the new approach within a laboratory air-conditioning climate control process and implemented in an industrial PCS as described in the *eighth chapter*.
- Finally a general discussion and a view to further work are reasoned in the *ninth chapter*.

2 Review of Approaches to Computer Aided Control System Design (CACSD)

Potentially fruitful interactions between the control and computer science communities in industry and academia are not properly handled today. According to Benveniste and Aström (1993) this was one of the main motivations to start the IEEE project "Facing and Challenge of Computer Science in the Industrial Applications of Control" and it impressively shows the need for applied research. Clearly, this literature review cannot address all aspects of CACSD systems but it will show the need for practical control system design as part of an industrially oriented CACSD procedure.

2.1 Current Control System Design Practise in Industry

Nowadays approximately 95% of all industrial controllers in process industry are PID-based (Nöth and Keuchel, 1996). This popularity of PID control is due to its simplicity and transparency. Experienced commissioners tune PID controller on singlevariable processes on the basis of their experience and by rules of thumb (Schuler, 1992). The process is implicitly regarded as a second order system which is not explicitly modelled. Another advantage is that rules and tuning aids are provided resulting in a well appreciated performance to cost ratio (Richalet, 1993). Well established simple parameterisation techniques like Chien-Hrones-Reswick or Ziegler-Nichols (Piwinger, 1975) are mostly based on simple step response experiments and are easily comprehensible. Also robust tuning of PID controllers based on step responses is possible as described by Maffezzoni and Rocco (1995) but this technique is rather advanced and rarely applied. Increased ease of use of PID controllers can be achieved by the implementation of automatic tuning and adaptation techniques which were introduced in the eighties, Aström *et al.* (1993) offer a comprehensive insight into this particular subject. They stress that automatic tuning (being mostly based on step response identification as well) is quite helpful for industrial users to build up gain schedules and to initialise adaptive controllers, thus providing the base for the control of nonlinear SISO processes.

Aström and Häggelund (1995) summarised two papers (Bialkowski, 1994; Ender, 1993) documenting their experience on the quality of industrial control. The result is surprising. Only 20% of the control loops fullfil their duties. The reasons are that in 30% of cases the specifications of sensors and actuators are not suitable, in another 30% the control performance is bad due to inappropriate controller tuning and the last 20% have other reasons. In most cases major malfunctions could have been avoided, if process and control engineers could have worked closely together in order to detect process design faults or equipment unsuitable for control. However, this is not the practise generally.

In the case of nonlinear multivariable processes the control performance of multiple singlevariable controllers is often inadequate, even though this is the standard control design in process industry. With the tendency to improve the performance of industrial processes, for example through the feedback of resources (such as heat and (by-)products), couplings are introduced so that manual readjustments of poorly designed controllers become necessary, often during operation. This reduces the product quality and can lead to unsafe process operation. Extra process staff are also required to undertake the frequent adjustments (Hahn and Nöth, 1997). Poor control performance can lead to critical situations especially in the case of process disturbances, such that the process personnel need to react quickly to switch off the controller and to manually adjust the process. More complicated control schemes have been designed only for a few multivariable processes, which are widely based on PID control but often comprise extra linear and also nonlinear elements like multipliers and min/max selection, which make the control system difficult to analyse (Tham *et al.*, 1991). Once the control strategy has been successful to a sufficient extent, these schemes become standard and are supplemented only if the final performance test on the process fails. However, if these standard control schemes are not available or applicable, most companies run their processes with simple sub-optimal control strategies because they are deterred by the enormous cost of modelling (Froese, 1995). Of course, such controlled processes need frequent supervision, in case of disturbances or drift. The frequent use of sub-optimal control strategies – neglecting the wide range of control methods that modern control theory provides – is generally justified by the predicted effort needed to improve the control performance (Funk, 1994).

2.2 Control Design Effort versus Performance

Today a vast variety of academic papers on control system design methods are published and discussed (see e.g. 15th World Congress of IFAC proceedings, 2002). There are a lot of advanced control system design methods, like state-space control, model predictive control, self-tuning or H_∞ control, most of them require high control expertise and tuning effort (Linko and Linko, 1998). When focusing on process industry approximately 95% of all industrial controllers are PID-based (Hahn and Nöth, 1993). However the two authors stressed also that commissioners and process engineers estimate that 50% of them can be optimised considerably. Commissioners have the difficult job of integrating automation and process functionality in a minimum of time. Most of them use only simple controller tuning rules of thumb like the Ziegler-Nichols method or they generate parameterised process models using step response (Nöth, 1998). Even these tools are not used very often and the controller remains poorly tuned as no process model is available. Many control system design strategies utilise singlevariable (SISO) control schemes (Aström and Häggelund, 1984) but in the case of nonlinear and multivariable (MIMO) processes the control performance of SISO control schemes are often insufficient. Only for a few special MIMO processes have more complicated standard control schemes been designed (e.g. Zhuang and Atherton, 1994; Tham *et al.*, 1991), which are widely based on PID control but comprise extra linear and also nonlinear elements like multipliers and min/max selection. For the development of automation strategies for MIMO processes it is natural for industrial engineers to split the process into independent SISO subprocesses. Then for each SISO subprocess an individual PID controller is designed often from rough estimates of dominant time constant and gain of the investigated subprocess. Final control parameterisation is done by rules of thumb or by simple manual tuning of the process. Only if this simple approach fails will the controller design procedure be extended to reduce nonlinear or coupling effects.

If the unused optimisation potential is to be utilised the effort that is needed to reach a better control performance has to be reasonably small. The comparison between design effort and control performance of PID control with different techniques, shown in Figure 2-1 (after Perne, 1995), helps to illustrate the problem. Usually installation and commissioning effort for a standard PID controller is small and the resulting control performance is sufficient in many cases in process industry. If better performance is required, other techniques have to be applied.

'Advanced PID' control for example is based on the same PID-algorithm but includes simple parameter adaptation, disturbance cancellation or the possibility to extend the control structure by different linear or nonlinear elements. These features would be extremely costly if realised by analogue techniques. Most digital industrial controllers offer the possibility of 'Advanced PID' control to increase the control performance, but these extras are rarely used. This is mainly because appropriate control system design methods based on process models and systematic controller design are not normally accessible to the non-expert users (Hahn and Nöth, 1993). With respect to multivariable systems Rake and Enning (1996) criticise the fact that no single industrial controller supports the identification of coupled systems and the corresponding PID control system design.

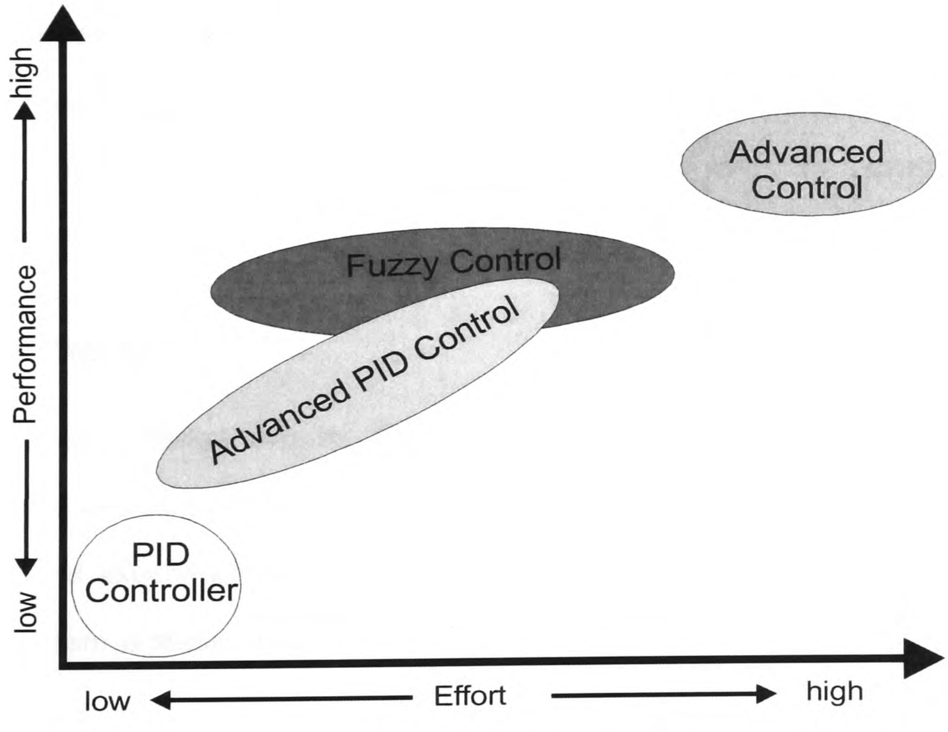


Figure 2-1 Performance and Effort of different control design techniques (after Perne, 1995)

'Advanced control' standards for the high end in control system design, like state-space control and model based predictive control (MPC), most of which require a great deal of computational effort and expertise in tuning and are therefore not suitable for everyday industrial control. According to Funk (1994) approximately five percent of all control loops in process industry are based on advanced control strategies with the proportion tending to increase.

'Fuzzy control' system design differs notably from the aforementioned methods incorporating rule-based process knowledge. It can show good performance with little control system design effort, if process knowledge is already available, but otherwise it needs significant optimisation effort to obtain a control performance, which possibly could have been gained through 'advanced PID' control more easily.

Comparing the different approaches 'advanced PID' control is the only control system design technique that displays a gradient in the performance-effort-area, implying that more effort in control system design leads to better control performance.

2.3 CACSD Systems

Control system design can be a highly complex and an intellectually difficult engineering design task. Generally speaking, the designer aims to use his skills, intuition and experience to design a controller such that the overall closed-loop control system is stable, robust and has good performance characteristic (e.g. fast response time and rapid settling time) (Hahn and Nöth, 1993). In order to carry out the task efficiently, the designer may rely on some Computer Aided Control System Design (CACSD) packages to assist him. These packages allow the designer to implement and test the design algorithms as well as to generate various kinds of plots or diagrams for the design analysis purpose.

More than 50 CACSD packages are available for PCs (Schumann, 1998) and an equal number of workstation versions (Schmid, 1993) are on the market, so it is very difficult for the interested application engineer to select an appropriate tool. Many researchers are active in the use of expert system techniques in control (Aström *et al.*, 1992; Pang, 1994). Schmid and Schumann (1993) stressed the importance of defining a standard exchange format for CACSD data because the number of different data formats is almost as high as the numbers of CACSD packages. King and Gray (1986) and Taylor and Frederic (1984) discussed the necessity of an object oriented architecture for CACSD tools. Barker (1988) and van den Boom (1988) emphasised the importance of the development of graphical tools for CACSD. Barker *et al.* (1990) concluded that the graphical tool should be the basis of a coherent working environment, which can support several CACSD software packages and should be especially tailored to the requirements of industrial users.

2.3.1 Use of CACSD Systems in Industry

It is surprising that the academic literature presented above tends to neglect the practical aspects which are important for the application of control system design in industry. The available CACSD systems have been developed for control experts, most of them in an academic environment, and serve as testbeds for new control methods providing many degrees of freedom for the tuning of these methods. These academic CACSD systems are not suitable for industrial users because the user often needs to switch between several CACSD programs to execute all the required CACSD tasks and because of the complexity that stems from their theoretical control system design methods which have large numbers of free parameters. Using academic CACSD programs for controller design for SISO, nonlinear or even MIMO processes will lead in general to mathematically complex process models and to the use of powerful theoretical controller design methods which are understandable and handled well only by control experts. This seems true even if the CACSD program is equipped with a sophisticated user guidance system as described in (Meier zu Farwig and Unbehauen, 1991). Taylor and Chan (1999) described the situation for industrial users as one where they are buying a system that seems to provide a lot of unnecessary functionality or else there is a frustration that they can not take advantage of functionality that they believe they need but can not use effectively.

Körner and Schumann (1996) suggest that a prerequisite for the industrial use of CACSD systems would be that all control system design tasks should have efficient support, reliable design results and could be reproduced by the industrial user and that the resulting control systems should be understandable and suitable for industrial needs.

The industrial user needs one integrated tool which is based on a block oriented simulation environment (Körner, 1996), to take care of all CACSD tasks including data acquisition, process identification, control design, optimisation, prototype control and control structure implementation. It should be streamlined to support the industrial users without requiring a high degree of theoretical control knowledge (Schumann *et al.*, 1996). CACSD designers and tool makers should be finding ways to break down rather than to build-up the complexity of the packages that are used for industrial computer system design tasks (Jobling, 1996).

MacFarlane *et al.* (1989) described the ideal CACSD system so comprehensively that the following passage became a main guideline for the development of CACSD systems. These should enable the designer to

1) "build; analyse; browse; search; compare and evaluate; reason and hypothesise; synthesise; design; manipulate and modify; experiment; catalogue; store and retrieve"

while being

2) "easily comprehensible to a single individual; wide in scope; modular with a manageable number of distinct parts; predictable in its behaviour; integrated and coherent in the way in which its different parts relate; helpful, with quick and efficient access to relevant information; tolerant of errors and supportive in enabling the effect of errors to be easily undone; extensible and adaptable; self documenting."

Naturally, not all of these objectives have been achieved yet. While most properties specified in 1) could be realised in many CACSD systems to different degrees, those specified in 2) have not been achieved fully, although some efforts have been undertaken.

For efficient use in industry two main prerequisites must be satisfied by CACSD systems:

- The effort needed to reach better control performance must be reasonably small.
- The resulting control technique must be transparent and trustworthy.

The first point is directly aimed at the user-friendliness of the CACSD system and its ability to support even inexperienced users doing good control design. This includes the critical modelling phase that must be intelligently supported as well. However, if the solution is not transparent for the industrial user who has to take account of proper process operation, even 'good academic control design' might fail in practise because of missing confidence. Hence it must be questioned if industrial users in process industry really need a CACSD system that supplies an overwhelming functionality as provided by most CACSD systems today. Some explanations for this situation can be found in the historical development of CACSD, which has been carried out mostly by academic control communities.

2.3.2 Milestones in the History of CACSD Systems

A short summary is given here about the historical development of CACSD highlighting the most important developments that improved the usability of CACSD systems as shown in Figure 2-2.

Program Libraries

In all probability the automatic synthesis program (ASP) was the first CACSD program designed by Kalmann and Englar (1966) to calculate optimal linear state space feed back systems. In the 1970s many control program libraries were developed, starting in 1971 with VASP (Variable Dimension Automatic Synthesis Program). EIPACK (Garbow *et al.*, 1977), Linpack (Dongarra *et al.*, 1979) and LAPACK (Anderson *et al.*, 1995) followed. Nearly all libraries were written in FORTRAN, being quite suitable for the mathematically oriented needs of the researchers in the 70´s and 80´s. Control program libraries worked successfully for testing, spreading wide and discussing the collected tools with interested scientists.

CACSD Packages

Based on these reliable libraries, CACSD packages like MATLAB, $MATRIX_X$ and Ctrl-C have been developed as interpreter interfaces to the numerical linear algebra libraries. With regard to non-linear systems, $MATRIX_X$ and Ctrl-C were first with the block-oriented modelling extensions, SystemBuild and Model-C respectively. Later on, MATLAB followed with SIMULAB (now: SIMULINK). For more details on these developments please refer to Taylor *et al.* (1991). However, the inherent suboptimal numerical performance caused by interpreted execution of the above mentioned commercial programs and restricted programming semantics, triggered in the mid eighties the development of heterogeneous packages. These packages are also called CACE (Computer Aided Control Engineering) systems. Packages like CADACS (Schmid, 1993) or ANDECS (Grübel, 1995) are good examples of this trend comprising functions for identification, system analysis, model transformation, controller synthesis, optimisation and sometimes even online facilities.

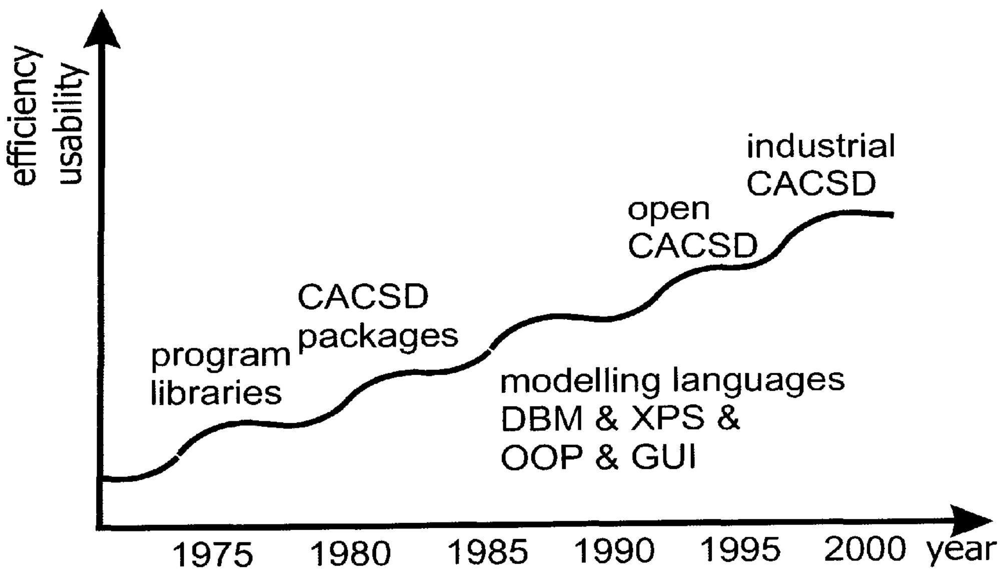


Figure 2-2 Impacts on the CACSD development

Efforts to Handle the CACSD Complexity

In 1986, IFAC set-up the working group on Guidelines for CACSD software. The IEEE Control System Society (IEEE-CSS) established a technical committee on CACSD. Their combined efforts have focused on three areas: user interfaces, data structures and algorithms. For an overview of the state of CACSD over this period, see the IFAC Proceedings of CADCS ´88 (Zhen-Yu, 1988) and of CADCS´91 (Barker, 1991). As the iterative control system design methodology generates a great deal of information, database management tools and expert systems dominated the CACSD development in the late eighties in order to make the use of CACSD systems manageable.

Modelling Languages

With respect to modelling it is now widely recognised that object-oriented modelling languages particularly will play an important role in CACSD future developments (Mattsson *et al.*, 1993). Equation based modelling languages like Dymola (Elmqvist, 1994), Omola (Anderson, 1989) or the currently developed unified modelling languages Modelica (Elmqvist *et al.*, 1998) provide concepts for model structuring to support multidisciplinary modelling and to facilitate reuse. However, these developments are only interesting for industrial users if appropriate user interfaces are available that allow their use in ways apart from the equation based level. Naturally it would be advantageous for extensive industrial use, if one of these languages would at least become the de facto standard.

Object-Oriented Programming

Nowadays one of the most popular ways to realise software, including the case of CACSD, is the application of object-oriented methodologies. It provides the potential to increase the software productivity and decrease the maintenance effort while reflecting natural concepts as comprehensively described by Jobling *et al.* (1994). Object-oriented programming (OOP) is useful to realise graphical user interfaces (GUI) and also to realise concepts like Open CACSD.

Open CACSD Development

A topic of special importance is the standardisation and development of 'open software' for CACSD that is easy to interface and maintain. In panel discussions on Open CACSD several routes to Open CACSD have been discussed (Taylor *et al.*,1994). Herewith DSblock (Otter and Mostermann, 1998) was recommended as an independent model bus in order to interface separate modelling and simulation environments (Grübel and Jobling, 1994). To realise the Open CACSD paradigms Barker *et al.* (1993) proposed a scheme that borrows ideas heavily from computer aided software engineering standardisation efforts. Schmid and Schumann (1994) presented a data interface standard for CACSD resulting from the work in the GMA (German measurement and automation society, 'Gesellschaft für Mess- und Automatisierungstechnik'). Lately, Barker *et al.* (1996) proposed an object-oriented approach to project management in CACSD incorporating many valuable ideas concerning multiple users, hierarchical systems, restrictions on model access, documentation and supervisory mechanisms. They bedded the model in a hierarchical block schematic diagram, which was described in the same issue by Varsamides *et al.* (1996), who introduced a very promising object-oriented information model and suggested a possible architecture for its implementation. Nevertheless they ended their paper very realistically: "However, to actually make use of this (or other similar) models, a wider consensus about fundamental aspects, like that of the CACE information of various levels, must be first achieved". This clearly shows that the results of these efforts would have no major effect on commercial packages as long as there is no consensus. Thus there has been a quest for de facto standards in CACSD that profit from their general acceptance in universities and industry.

2.3.3 MATLAB – De Facto Standard in CACSD

During the last decade more and more MATLAB based toolboxes have been available supporting special aspects of scientific problems. These MATLAB toolboxes have been especially very successful because of their simple and flexible extendibility via C-like m-files (MATLAB-files). Rimvall (1988) detailed the toolbox concept and showed possible extensions. MATLAB has also been winning when compared to other packages like MATRIX_X and Ctrl-C because of the synergetic relationship between the signal processing, system identification and control system toolbox which cover many CACSD problems (Sørlie, 1994). Hence it is possible to develop packages in this proprietary environment utilising a large and well established software base.

Besides, MATLAB´s user interface is especially geared to researchers in control, who prefer to communicate with computers using numerical data models such as state space or polynomials, which are traditionally represented by arrays. Many scientific publications incorporate MATLAB simulation examples in the field of control and researchers often exchange their ideas in form of m–files. MATLAB´s dominance in the field of CACSD is also demonstrated within the extended list of control software ELCS (Frederick *et al.*, 1992), where all MATLAB compatible programs are summarised in an extra section ´MATLAB-family software packages´ because of their large number and relative importance.

MATLAB also provides the block-oriented simulation environment SIMULINK, which is increasingly accepted in industry and which can handle linear and nonlinear, continuous and discrete models as well as combinations of them (Hortsch and Schlüter, 1996). Nevertheless MATLAB is not geared to the Open CACSD efforts as outlined above, but it provides an OLE interface allowing execution of MATLAB functions from external programs and the ability to generate C-code from SIMULINK block diagrams. In the context of modelling languages it is possible to link the widely used physical modelling tool Dymola to SIMULINK, for example for use with controllers implemented in SIMULINK.

2.4 Modelling as a Crucial Part of the CACSD

In a study about the industrial application of model based predictive control modelling Richalet (1993) assessed that modelling, experiments and identification require more than half the effort needed for advanced control system design in process industry. Foss *et al.* (1998) also detected modelling as 'the main bottleneck for the application of advanced control' and investigated the requirements for computer-based tools supporting the modelling process. They performed a field study of the industrial modelling process, interviewing experienced modellers. However useful this approach is, it still focuses on the demands of specialists. Takutsu *et al.* (1998) reported on the future needs for the Japanese control industry. They detected that the lack of the process analysis is the key factor for failure of automation strategies and that modelling tools are still often seen as the most lacking engineering environment. Therefore it is necessary to investigate strategies that support process modelling, especially for industrial users.

2.4.1 Process Modelling and Identification

A model is 'a representation of the essential aspects of an existing system (or a system to be constructed) which presents the knowledge of that system in a useable form' (Eykhoff, 1974). For the solution of many scientific and technological problems mathematical process models are required, which imitate the most interesting properties of the process.

The two basic approaches to modelling are theoretical and experimental modelling

- Theoretical modelling is based on the underlying physical laws and offers an excellent insight into the process behaviour. It utilises abstraction, decomposition and aggregation of sub-models describing the process partly to compose a structured process model (Cellier, 1991), which also provides information about the internal state of the process. Therefore the evolving process models are called white or transparent box models.
- By contrast experimental modelling – widely called identification – is concerned with the generation of mathematical process models from experimental data. Naturally, the process must already exist and it must be regarded that the identified process model is only valid within the analysed range and does not necessarily comprise physically relevant parameters. Therefore these inductively derived models are called black box models or grey box models, when the structure of the process model is known, as are parameters, but their values are unknown and have to be identified. Due to rapid and context related results identification is an efficient method for modelling aimed at control design, especially during commissioning. However, as the process is already built, sensible changes in process design are expensive or even impossible.

While the knowledge of the internal process structure has to be profound for the theoretical modelling in order to yield satisfactory model performance, less knowledge is needed for the application of identification methods to obtain models enabling satisfactory control to be achieved.

In this context three types of processes can be distinguished with respect to the knowledge available of the internal process structure (Isermann, 1991):

1. good knowledge (for example mechanical and electrical processes)
2. less knowledge (for example processes in power engineering)
3. poor knowledge (for example processes in chemical or process engineering)

Especially for processes of type 2 and 3 it is sensible to apply identification techniques. Furthermore in practical application identification methods can be applied without requiring too much effort and expertise and thus cost.

2.5 Conclusion of this Chapter

The review of CACSD systems has provided a taste of the immense amount of research carried out in this field. However, the transfer of the scientific results into industrial application is rather slow, a situation which can also be found in other scientific fields and which is visualised in Figure 2-3.

Currently industry faces a paradoxical situation:

The more advanced automatic control becomes, the more advanced must be the system engineer to apply it in practise.

This should not necessarily be the case. Occasionally the computer aided design (CAD) support of industrial issues is already quite good and straightforward in application, even for inexperienced users. However, the necessary precondition is an appropriate control system design.

The requirements for an industrial identification tool which were formulated by Körner and Schumann (1998) and realised in the MATLAB™ toolbox *ICAI* (Körner, 1999), are similar to the requirements of the industrial controller design procedure for the reliable support of inexperienced users during the control system design procedure. The main requirements are:

- Integration of the control system design task into a block oriented simulation environment, such that the user does not have to use different tools for the individual design tasks of identification, control system design and simulation between different programs.
- Only a few and robust control system design procedures should be available.
- Support of complex control system design tasks up to nonlinear MIMO control through standardised procedures.

and additionally

- Support of PID-based control systems which can be implemented by industrial process control systems (PCS).

3 Review of Control System Design in Industry

This chapter gives a review of control methods used in industry, and methods and technologies for industrial control system design. Attention was paid to those methods and technologies of modern control system design that may be suitable for industrial practise. This review cannot address all aspects of control system design but it will show the need for practically applicable control system design methods for nonlinear MIMO processes as part of an industrially oriented CACSD procedure supporting inexperienced users. At the end of each section a brief description of available tools is given.

3.1 PID Control

Although many innovative methodologies have been devised in the past 50 years to handle more complex control problems and to achieve better performances, the great majority of industrial processes are still controlled by means of simple proportional-integral-derivative (PID) controllers (estimates range up to 95%) (Yu, 1999). This seems to be the case because PID controllers, despite their simple structure, assure acceptable performances for a wide range of industrial plants and their usage (the tuning of their parameters) is well known among industrial operators (Gagnon, 1999). Hence, PID controllers provide, in industrial environments, a cost/benefit performance that is difficult to beat with other kinds of controllers. Although, the proportional-integral-derivative (PID) controller has only three parameters, it is not easy, without a systematic procedure, to find good settings for them. In fact, a visit to a process plant will usually show that a large number of the PID controllers are poorly tuned (Skogsted, 2003). Therefore the developments in this area are very interesting, as they affect industrial practice.

Linear SISO PID Control

Recently, it has been noticed that PID controllers are often poorly tuned (Visioli, 2004) and some efforts have been made to systematically resolve this matter including the classical paper by Ziegler and Nichols (1942). The Ziegler-Nichols methods were designed to give good responses to load disturbances. This is not satisfactory for many systems. Additionally, Ziegler-Nichols methods are not easy to apply in their original form on working plants. When critical processes are involved, sudden changes in the control signal or operation at the stability limit are often not acceptable. Relay feedback and describing function analysis (Aström and Hägglund, 1988) are often applied for parameter identification to overcome the above problems. Visioli (2001 b) has proposed tuning methods for integrating systems based on minimising ISE (Integral Squared Error) and ITSE (Integral time Square Error). The optimisation is carried out by a genetic algorithm.

To improve the control performance, several schemes of self-tuning PID controllers have been proposed in the past. Wittenberg (1979) proposed a control structure with the PID algorithm calculated via pole placement design. The method was limited in the order of controlled processes. The self tuning PI or PID algorithms were automatically derived from the dynamic of the controlled process (Grawthrop, 1986). An alterative self-tuning PID controller was proposed by Clark and Gawthrop (1979) and Cameron and Seborg (1983) and based on the generalised minimum variance control. The control structure was oriented to have a PID structure. The controller parameters were obtained using a parameter estimation scheme. A brief summary of PID theory and most tuning methods are discussed by Cominos and Munro (2002). Other forms of PID controller setting methods can be found in the literature (Chidambaram and Sree, 2003, Skogestad, 2003) . However, these tuning and self tuning adaptive control approaches are limited to linear system theory.

Nonlinear SISO PID Control

In nonlinear control design theory, linearisation of nonlinear models is often used to facilitate the design of controllers for nonlinear systems (Chen and Huang, 2004). This linearisation around the operating points is a relatively inaccurate approach (Zhu and Warwick, 2000). Tayler and Aström (1986) introduced a nonlinear PID auto-tuning algorithm which combines the Aström/Häggelung algorithm for linear cases with the approach to nonlinear compensator synthesis of Taylor/Strobel. During the last years model predictive control, fuzzy or neural networks have been used for the nonlinear PID control design and will be described in the following sections.

MIMO PID control

Control of complex industrial processes like MIMO processes are not well resolved with a classical PID controller (Zamarreno, 2002). It is more difficult to design PID controllers for a MIMO process than for a SISO process because there is an interaction between the different control loops in a MIMO system which affects the others performance. The tuning of MIMO systems with finite frequency response data was introduced by Rosenbrock (1974) with the optimisation of diagonal PID controllers with off-diagonal decoupling gains. Hang *et al.* (1991) described a method for auto-tuning of multivariable decoupling controllers. The decouplers, used to cancel the effect of the coupling between loops, were designed with the identified transfer functions of the main and coupling loops. The decoupled main loops free from interaction are then tuned using conventional PID tuning rules like

Ziegler-Nichols. An auto-tuning procedure to determine a diagonal PID controller by using the so called generalised Ziegler-Nichols method for a two-input two-output (TITO) system, was also introduced by Zhuang and Atherton (1994). An approach to design a multivariable PID controller by linear quadratic state space was described by Hensel and Peter (1985). This method evaluates discrete-time PID parameters for multivariable processes by designing a state space controller with integral action and approximation of its control behaviour to a PID controller.

All these methods from linear SISO to MIMO controller design are the result of academic research and an inexperienced industrial user can not easily get information of the control systems. The theoretical methods of the process control systems can only be found in a few industrial applications. The SISO PID controller is still the favourite choice for industrial engineers. Only for a few complex processes are more advanced methods are successfully used (Schuler, 2003). The design of PID-based control systems for nonlinear multivariable processes are still missing.

3.1.1 Industrial Tools for PID Controller Design and Assessment

At trade fairs it is clear that almost all producers of digital compact controllers and process control systems offer functionality for self tuning . Although most producers hold back publication of detailed information about the methods they used, they usually derive from two essential methods (Aström and Häggelund, 1995). The most wide spread method is the evaluation of the step response of the controlled system (tangent of the reversing point and its modification). The behaviour is presupposed to be aperiodic. In a small number of controllers the auto-tuning method from Aström and Häggelund (1984) is used, which uses a two-step controller in the closed loop. The amplitude and the time of oscillation of the continuous oscillation which arise are fixed and give a recommendation for the parameter settings of the PID controller. All these self tuning methods are rarely used in industrial practice (Rake and Enning, 1996). For the use of more demanding design methods (for example advanced PID controllers), it is necessary to use efficient identification methods for generating process models. Most of the controller producers also offer the possibility of automatically changing the PID control parameters depending on different variables (for example set point or control deviation), often also called parameter scheduling. This simple form of adaptation in the open loop is sometimes used in practise, for example to control processes with a well-known nonlinear static behaviour or for averaging level control. As Reinig *et al.* (2000) discussed, the continuous assistance of process identification, control scheme, simulation and implementation is still missing.

Some tools have been developed for the evaluation of controllers (controller assessment). These tools can on the one hand list, archive and analyse operating conditions of the controller (e.g. hand or automatic operation) and on the other hand value the control performance including comparison with the best performance. Thereby the Control Performance Index proposed by Harris *et al.*, (1999), compares the real variance of the control variable with the variance of an unrestricted minimal variance controller. This index is widely accepted in industry (Reining *et al.*, 2000). Table 3-1 gives an overview of some products for the assessment of controllers.

Table 3-1 Overview of products for controller assessment

Product	Peculiarity
AspenWatch	reception, archives and evaluation of DMCplus application incl. Diagnostic collaboration with AspenTech´s Sustained Performance program
DeltaV Inspect	DeltaV – standard component evaluation of all relevant signals and control (CPI and other statistics) alarm
Loop Optimizer Suite	control loop optimisation (with LoopOptimizer) offline evaluation of the controller with Loop Audit Evaluator intermeshed controllers are developed personnel training with Loop Professional Trainer
ProfitLoop (LoopScout)	evaluation of the control (CPI and other statistics) diagnostic of the system components proposal for changes of tuning and structure by experts

3.2 Model Predictive Control

Model predictive control (MPC), probably more than any other advanced control technique, has rapidly found significant application in the process industry. The technique was initiated and developed within industry, principally by engineers at Shell Oil, in the early 1970s. Now, 30 years later, the method is reaching maturity with a thorough analytical basis and successful applications. Nowadays these methods are utilised successfully in many other areas like chemical, paper, cellulose and food industries and also power plants. Therefore it is obvious that the MPC method has a special relevance under the advanced control design methods (Qin and Badgewell, 1996; Takatsu *et al.,* 1998; Schuler and Holl, 1998). Qin and

Badgewell (1996) stated that the total number of reported applications is greater than 2200 and Reining *et al.* (2000) estimate around 3000.

Some essential reasons for the success of model predictive control tools are that (Reining *et al.*, 2000)

- they are useful for processes with complex dynamics because of their model based function,
- they support the consideration of restrictions from the manipulated variables and enable the optional demand both from the set point and the range of the control variable,
- they raise the possibility of handling non symmetric multivariable processes with temporarily variable structure,
- in the case of nonlinear models the effort for the modelling of the dynamic behaviour is reasonably high,
- in the meantime an efficient service with supporting companies exists and
- the support of the control systems after the commissioning can be managed by non specialists.

Several reviews and comparative studies of the main MPC algorithms have been published, for example de Keyser *et al.* (1988), Garcia *et al.* (1989), Kramer and Unbehauen (1991), and Qin and Badgwell (1996). Their application in particular to the chemical process industry has been described by Eaten and Rawlings (1992). Industrial applications have also been described by Richalet (1993). There are several textbooks on MPC available like Camacho and Bordons (1999).

The various techniques are distinguished by different types of model and performance function employed. Presented from an historical perspective, some of the main algorithms are:

- Model algorithm control (MAC) (Richalet *et al.*, 1978), initially called model predictive heuristic control (MPHC). This uses an impulse response model, which is valid only for open loop stable processes, and minimises the variance of the error between the output and the reference trajectory computed as a first order system.
- Dynamic matrix control (DMC) (Culter and Ramerker, 1980). This is similar to MAC but uses a step response model instead of an impulse response model. This technique was originally applied in Shell Oil as early as 1973. The method was extended to include input and output constraint handling using quadratic programming to solve the constrained optimisation problem, giving rise to quadratic dynamic matrix control (QDMC) (Cutler *et al.*, 1983). The DMC algorithm can also be derived for a general discrete state space model (Prett and Garcia, 1988)
- Extended prediction self adaptive control (EPSAC) (de Keyser and van Cauwenberghe, 1985) uses a discrete (z-transform) transfer function to model the process and a simple control law structure calculated analytically using the quadratic performance function assuming $u(t)$ stays constant from instant t. The process model also includes measurable disturbances.
- Generalised predictive control (GPC) (Clarke *et al.*, 1987), uses a quadratic performance function with weighting of the control effort, and an ARMAX model (auto-regressive moving average with exogenous variables). It also

provides an analytic solution for the optimal control in absence of constraints.

DMC and GPC are perhaps the most popular techniques. There are several extensions to GPC including techniques which guarantee stability through end point equality constraints (Clarke and Scattolini, 1992), and by stabilising the process prior to the objective function optimisation (Kouvaritakis *et al.*, 1992). The GPC technique, using a method known as constrained stable generalised predictive control (CSGPC) (Rossiter and Kouvaritakis, 1993), has also been extended to guarantee feasibility and stability when there are input constraints as well as terminal constraints (Rossiter *et al.*, 1996). Although most of the work has been performed in discrete time, GPC has also been formulated in continuous time (Demircioglu and Gawthrop, 1991). A state space model description for GPC controller has been developed by Ordys and Clarke (1993).

The design and combining multiple linear DMC controller was proposed by Dougherty and Cooper (2003). The technique requires that each controller has their own step response model that describes the process dynamic at a specific operation region. The adaptive DMC controller is able to maintain consistent set point tracking performance over the range of nonlinear operation. A nonlinear generalised predictive control (NGPC) approach for a nonlinear PID controller was derived by Chen *et al.* (1999). For a nonlinear system with a low relative degree proposed this approach that the optimal NGPC has a PID structure. In addition to the PID structure the controller has a prediction part. It can predict the trend of the system's output using the knowledge of the system and the current states and thus compensate it. Another design procedure was proposed by Pomerleau *et al.* (2003) which considers relevant process characteristics, such as nonlinearity, for the synthesis of decentralised extended PIDs and model-based predictive controllers. The control strategies proposed consists of a linear controller with an inverse of the static nonlinearity.

At present, predictive control research seems to focus on model-based predictive control, PID types of predictive control, fuzzy logic based predictive control, and neural-net predictive control. In order to meet the increasing demands on quality and economic performance of control systems, predictive control is gaining attention gradually . Vega and Prada (1991) described a self tuning method of predictive PID controllers, which are becoming a promising branch of predictive control research today (Brown, 1993). Through a combination of predictive and fuzzy PID control Lu *et al.* (2001) described a method to form a new predictive fuzzy PID controller. An industrial case study of the performance evaluation of an industrial MPC controller implemented on a 6-square industrial process was presented by Gao *et al.* (2002).

3.2.1 Tools for Linear Model Predictive Control

The majority of industrial applications were and will be realised by MPC technology based on linear process models (LMPC). Table 3-1 gives an overview of some LMPC packages.

This overview shows that all program packages have the same structure: on the one hand they include tools for process identification based on measured process data, tools for controller design and tools for offline simulation of the closed loop. On the other hand there is an online version of the MPC algorithm which uses the identified dynamic process model. The online MPC controller is implemented in special hardware and is connected by an appropriate real-time interface to variables of the process control system (PCS). Some manufacturers (e.g. AspenTech, Honeywell, SIMSCI) also have the possibility to generate cascade controllers to support process overlapping control tasks.

Table 3-2 MPC-Techniques with linear process models

Product	Software Functionality	Remarks
Connoisseur	Connoisseur Plant Analysis System (identification and generating models, ANN-models, controller design and offline simulation)	- interfaces PCS from different suppliers with special developed interfaces, OPC-Interface under development - interfaces dynamic simulators like DynSim, Speed-UP and MATLAB
DMCplus	DMCplus Model (identification and model generation) DMCplus Builder (MPC controller design) DMCplus Simulate (offline simulation)	- interfaces PCS from different suppliers with CIMIO- or OPC-Interface - interfaces Aspen´s Operator-Training-System and Engineering Suite for Dynamic Simulation
Profit Controller (RMPCT)	integration in Advanced Process Control Development Environment – APCEDE (identification and model generation, controller design and offline simulation)	- interfaces PCS from different suppliers, OPC-Interface under development - interfaces Hi-Spec´s Operator-Trainings-System TRAINER and to ShadowPlant under development
SMOC	SMOC Design and Simulation Software (AIDA – identification and generating models, SMOC-PC – controller design and offline simulation)	- interfaces PCS from different suppliers with OPC and PCS specific interfaces
3dMPC	3dMPC Engineering Suite (identification and model generation, controller design and offline simulation)	- interfaces PCS from different suppliers with OPC - interface to MATLAB and to the simulator ABB Process Dynamic

The special features of Connoisseur from *SIMSCI* are the options to adapt the dynamic process model in real time, to design the controller with fuzzy logic and the model generation through neural networks.

Aspen Technologies Inc. developed the MPC package *DMCplus*. *DMCplus* represents the combination of the well known predictive control algorithm (Dynamic Matrix Control) and the development area of the program system. The identification tool is called DMCplus Model and supports the experimental model generation, it generates for each subprocess, non parameterised FSR- (Finite Step Response) models. The DMCplus Build/Simulate supports the control system design and the off line simulation. The online version of DMCplus Controller can be connected with process control systems via OPC (OLE for process control) or specially developed interfaces by AspenTech.

Honeywell Hi-Spec Solutions offers the MPC package *Profit Controller* with the development area for Advanced Process Control Development Environment (APCDE). The identification tool Model Identifier supports the multi-level identification procedure. First, non parameterised models are estimated for every sub process and then these models are approximated with parameterised transfer function models. The Control Builder/Simulator supports the offline interactive design of the MPC system.

The *SMOC* package developed from Shell Global Solutions works like the previously discussed program packages. A special feature is the application of linear state space models to describe the dynamic behaviour of the process. Shell sees an application benefit in the improved recognition of the effects from non measurable disturbance variables in properly defined industrial processing variables of state (e.g. the temperature of the column on the control of the concentration in a stripping tower). The program AIDA (Advanced Identification and Data Analysis) is offered for process identification, a special feature of which is the multi-level identification procedure. First, non parametric models are estimated and then these models are approximated with state space models.

The MPC package *3dMPC* is a new product development from ABB Automation Products (3dMPC means "three dimensional design freedom" MPC). The results of recent research have contributed to the development of this package. Special features were mentioned: the execution of modern identification techniques, i.e. subspace methods for the identification of linear systems in state space (Larimore and Seborg, 1999), the possibility of independent settings for the control behaviour of reference action and disturbance reaction of the control loop as well as feedforward control and the interface for MATLAB/SIMULINK™.

3.2.2 Tools for Nonlinear Model Predictive Control

Practical experience shows that in some cases the control performance is not sufficient when using the MPC procedure with a linear process model. This is the case typically for processes with strong and pronounced non-linearity around the operating point (e.g. pH control) and processes with weak local non-linearity but which are operated in a wide area of operation (e.g. batch-processes in chemical or biotechnology industry). In those cases the application of Nonlinear Model Predictive Control (NMPC) can be a suitable method of solution (Qin and Badgwell, 2000; Henson and Seborg, 1997). The transfer from LMPC to NMPC control reveals many theoretical and practical problems, like identification of non-linear systems and the generation of models. The developments to solve these problems have their origin in the work of Allgöwer (1999). Nonlinear model predictive control based on state space models and the receding horizon concept has been developed (Balchen *et al.*, 1992), including stability analysis (Mayne and Michalski, 1990) and sub-optimal schemes that guarantee stability (Scokaert *et al.*, 1999). Integration of economic objectives within the performance function has been performed by Becerra *et al.* (1998). It is remarkable that some NMPC packages are available, and these are listed in Table 3-3.

Table 3-3 MPC-Techniques with nonlinear process models

Product	Development Area	Remarks
INCA	INCA Modeler, Test, View, Configurater, Simulator (identification and generating models, generation of test signals, controller design and offline simulation)	- interfaces PCS of different producer over OPC, TCP/IP and PCS specific interfaces - coupling to Gproms and MATLAB
Process Perfecter	ProcessInsights for generating models PFC Control Design for generating models and controller design	- interfaces PCS from different suppliers via Pavilion Data Interface (PDI)
Profit NLC	Profit NLC (Profit NonLinear Controller)	- theoretical static process model in the MPC controller - interfaces PCS from different suppliers, OPC Interface under development - cooperation with DOT Products (NOVA NLC)

The program system *INCA* (IPCOS Technology's Novel Control Architecture) from IPCOS and ISMC consists of the components INCAModeler (identification of linear multivariable systems). INCAConfigurator and INCASimulator (controller design and offline simulation), INCAView (operating and observing) and INCAEngine (online controller). Operational areas are power plants, as well as processes of polymer and glass industry.

The NMPC package *ProcessPerfecter* from Pavilion Technologies features the combination of non linear static models in the form of artificial neural networks and linear dynamic models. The applications of this package are especially multi product polymerisation processes.

Honeywell Hi-Spec Solution developed the NMPC package *Profit NLC* (Profit Nonlinear Control) for the operational area of multivariable polymerisation processes. This package utilises theoretical models for polymerisation processes which were developed and tested within a collaboration with Exxon.

MPC tools are very voluminous systems, the models are not particularly clear for nonlinear or even MIMO processes. Average industrial users are not familiar with the strategies and the handling of these tools. AspenTech admits that from the customers of the world-wide most commonly sold MPC package *DMCplus*, only 10% are able to use this very expensive engineering tool, other companies confirm similar numbers (Reinig *et al.,* 2000).

3.3 Fuzzy Logic and Fuzzy Control

Fuzzy logic and artificial neural network methods belong to the methods of artificial intelligence, which have gained in the last few years a significant role in the practise of process automation. The application area is wide spread and is not only utilised for process control. In fact these methods are also utilised for process monitoring, technical diagnostic procedures and process optimisation, amongst others.

Fuzzy logic provides methods to represent quantitative fuzzy definitions of the human speech and the ability of human beings to copy the handling of fuzzy in if-then-rules. This gives the possibility to embed human expertise into control. The practical experiences show that the application of fuzzy logic – often in combination with classical techniques - gives high performance when it is necessary to generate and handle nonlinear characteristic fields for the process control.

Also the fuzzy PID control method has been developed. This method has analytical structure and guaranteed stability, and shows good performance for controlling general nonlinear systems (Ying *et al.*, 1990; Chen, 1996). Fuzzy control is also used for controlling nonlinear systems whose plant models are unknown or vague. Combining it with predictive control may avoid model based GPC's drawbacks. Fuzzy predictive control was proposed by Choi *et al.* (1996) and Capriano and Seaz (1997). The combination of these two methods was proposed by Lu *et al.* (2003) to predictive fuzzy PID control, which is suitable for controlling uncertain linear and nonlinear systems with guaranteed stability. The ease of tuning of the fuzzy mechanism parameters plays a key role in the practical applicability of the methodologies, since it determines the improvement in the cost/benefit ratio with respect to standard methods (Visioli, 2001 a).

3.3.1 Tools of Fuzzy Control

The number of industrial tools is relatively small. The small overview in Table 3-4 shows a list of products for fuzzy logic and fuzzy control.

Table 3-4. Fuzzy Logic and Fuzzy Control

Product	Development Area	Remarks
DataEngine	different tools for data based system analysis including Fuzzy-Control and Fuzzy-Clustering	- DLL, C++ code generator - interfaces Visual Basic
FuzzyControl++	configuration tool FuzzyControl++	- maximal 8 inputs and 4 outputs
S40 Fuzzy Package	FuzzyTECH program system integrated in the user interface of Moeller PCS	- co-operation with INFORM GmbH Aachen (fuzzyTECH)
WinFACT 2000	modular build program system with tools for analyse, synthesis and simulation of the control system including Fuzzy-Logic-Shell FLOP and fuzzy PID controller	- co-operation with INFORM GmbH Aachen (integration of the fuzzyTECH package into WINFACT)
WINROSA DORA for Windows	WINROSA – automatically generation of fuzzy controller with measurement data DORA - program system with tools for analyse, synthesis and blockoriented simulation of the control system	- different novel fuzzy control structures - interfaces WINFACT, MATLAB and fuzzyTECH

DataEngine from MIT GmbH Aachen is a tool which integrates a multitude of methods for the intelligent data analysis within a general user interface. Besides statistical techniques and methods for signal processing it includes also fuzzy logic, fuzzy clustering and artificial neural networks. The tool DataEngine Application Development Library makes the DataEngine models suitable for other application programs (DLL, C code generator, Interface to Visual Basic).

Fisher Rosemount offers with *DeltaV Fuzzy* a fuzzy PID controller function block for the PCS ProVox and RS3. The control method and the linguistic terms are predetermined and the user can not influence them.

Siemens offers with *FuzzyControl++* a tool for PC and programmer that designates terms for linguistic variables which edit membership functions of the inputs and outputs graphically and tabulated rule base. Plots and prints of characteristic fields and dynamic representations facilitate the analysis of the control loop and the control performance. The online version of the controller can be implemented as function block into the PLS SIMATIC S7-300 and S7-400 and also as Smart Object within WinCC. With the interface to NeuroSystems neural fuzzy-systems can be realised.

The *S40* Fuzzy Package from Moeller makes the newest development of the program system fuzzyTECH suitable for the users of the PCS PS4.

The modular program system *WinFACT 2000* from Ingenieurbüro Dr. Kahnert includes on the one hand tools to analyse, design and simulate conventional control systems and on the other hand fuzzy systems and neural networks. WinFACT 2000 offers a fuzzy shell (Fuzzy Logic Operating Program, FLOP) for the development of fuzzy systems and a fuzzy PID controller. Beyond that it makes the use of other systems like fuzzyTECH and NeuroModel suitable within this system.

The Software *DORA for Windows* from the University of Dortmund has on the one hand a common analysis and control design method within a blockoriented simulator and on the other hand a fuzzy module which and includes not only standard methods but also new fuzzy controller structures. *WINROSA* is an additional tool which generates automatically static relevant fuzzy controllers by evaluating measurement data.

Fuzzy control systems are especially suitable for control structures which can be analysed and understood intuitively. Process knowledge is necessary to generate the rules. In the case of complex processes like MIMO processes or nonlinear processes a structured approach of the design models and rules are an uneasy task. Interest by industrial user in fuzzy logic for control design tools is very small (Reinig *et al.*, 2000) and potential costumers are ususally fuzzy experts (Rake and Enning, 1996).

3.4 Neural Networks

The technical application of artificial neural networks (ANN) is usually to nonlinear function approximations. Thereby the properties of ANN are utilized to map complex multidimensional nonlinear coherence with the principle of arbitrary exactness. The predominant application of ANN in automation practise is the development of so called soft sensors. These are model based measurement methods which train the ANN to predict quality parameters like concentrations of multiple material mixtures , densities and melt indices of polymers and form the basis of simple and cheap measurements (e.g. temperature, flow and pressure). Another application of ANN is the generation of static process models from the basis of historical process data with the target to realise a static operating point optimisation.

Currently neural networks enjoy a very large research interest. They have great capability in solving complex mathematical problems since they have been proven to approximate any continuous function as accurately as possible (Hornik *et al.*, 1989). Hence, it has received considerable attention in the field of chemical process control and has been applied to system identification and controller design (Bhat and McAvoy, 1990; Su and McAvoy, 1993). All of these works show that the neural network can capture the characteristics of system patterns and performance function approximation for nonlinear systems (Narendra and Parthasarathy, 1990). It is used in the control fields for modelling nonlinear processes in the model based control, such as a direct and indirect neural network model based control (Psichogios and Ungar, 1991), nonlinear internal model control (Nahas *et al.*, 1992). The control performance of the above methods are satisfactory, but the nonlinear iterative algorithm of the control design is computationally demanding because of the control system based on nonlinear neural network model. Combination of methods base on neural network and the traditional linear controller design have been developed. Fuli *et al.* (1997) proposed a comprised method with neural network and pole placement design. This method assumed that the plant could be linearised at each operation point. It used the linear neural network to capture the linear dynamic behaviour of processes, and the multilayered feedforward neural network to identify the nonlinear part as a measured disturbance. In the controller design, it used pole placement as feedback control, and the multilayered feedforward neural network as feedforward control to eliminate the nonlinear disturbance. This method is not concerned to processes with the PID controller. An on-line updated PID algorithm was proposed by Chen and Huang (2004). It uses a minimum variation control and a linearised neural network model around its current operation region.

3.4.1 Tools for the Development and Application of Neural Networks

Table 3-5 shows an overview of industrial tools for the development and application of neural networks.

Table 3-5. Tools for the development of artificial neural networks

Product	Development Area	Remarks
AspenIQ	AspenIQmodel for preprocessing, training and validation	- stand alone or in combination with DMCplus or InfoPlus 2.1
DataEngine	different tools for data based system analysis including artificial neutal networks	- DLL, C++ code generator - coupling to VisualBAsic
DeltaV Neural	PCS DeltaV	- online implementation as function block in DeltaV
NeuroIdentifier	identification of dynamic systems with neural networks of different structures, support of the structure choose	- stand alone product
NeuroSystems		- coupling with FuzzyControl++ for the developmet of neural fuzzy systems possible
ProcessInsights	different tools for formation and preprocessing of datas, training and analysis of ANN	- also part of the NMPC package ProcessPerfecter - coupling with PCS of different producer with Pavillion Data Interface
ProfitSensor	integration in Advanced Process Control Development Environment (APCDE)	- generation of linear models with MKQ, PLS and PCA
Robust Quality Estimator RQE	RQE offline package for preprocessing, training and validation	- model adaptation with Kalman filter techniques
MATLAB™ Neural Network Toolbx	Extension of the MATLAB computing environment to provide tools for the design, implementation, visualization, and simulation of neural networks	- Graphical user interface (GUI) - provides comprehensive support for many proven network paradigms

Aspen Technologies Inc. offers with *AspenIQ* a program system for the design and real time implementation of soft sensors. The design tool implies different methods for data preprocessing including the automatic temporal and correct appropriation of the data sets to select the relevant inputs and to generate different soft sensor models.

The program system *DataEngine* from MIT GmbH was already briefly introduced in Chapter 3.3.3. This tool for intelligent data analysis includes neural networks.

The development *DeltaV Neuro* from Fisher Rosemount is divided into an offline design tool (data preprocessing, net training, net analysis and validation) and into an online implementation. The online version is realised as a function block in PCS DeltaV.

The program system *ProcessInsights* from Pavillion Technologies includes a development area with many methods for data preprocessing. Worth mentioning is the application for the development of static process models with a new technique of nonlinear correlation analysis for the automatic temporal assignment of different process data under consideration of the process dynamic.

The program package for the development of soft sensors from Honeywell HI-Spec Solutions called *ProfitSensor*. This tool does not essentially differ from the packages discussed above, in its basic structure.

The offline tool *RQE Robust Quality Estimator* from Shell Global Solutions includes methods for generate linear models (Partial Least Squares, Principal Components Analysis) as well as for training neural networks.

Siemens offers *NeuroSystem* to facilitate the training of different networks, the results can be visualised in a multidimensional characteristic field graphic.

Application platforms are SIMATIC S7-300/400 (special function block) and WinCC (implementation as a smart object).

The *Neural Network Toolbox* extends the MATLAB™ computing environment to provide tools for the design, implementation, visualization, and simulation of neural networks. The Neural Network Toolbox provides comprehensive support for many proven network paradigms, as well as a graphical user interface that enables you to design and manage your networks. The toolbox's modular, open and extensible design simplifies the creation of customized functions and networks.

Neural networks are uniquely powerful tools that are used in applications where formal analysis would be difficult or impossible, such as pattern recognition and nonlinear system identification and control. Based on the previous discussion, the available neural network tools can be used to model nonlinear multivariable processes a basis for model predictive control systems, however the design task can only be handled by system experts, these tools are not suitable for an industrial user. Because of the very complex handling of these tools and the expert knowledge required for the design method.

3.5 Conclusion of this Chapter

The review has focused on controller design for nonlinear and multivariable processes, which are useable by industrial users without expert control knowledge. The reasons why these tools are not widespread in industry (Reinig *et al.,* 2000) are that despite the many systems available, insufficient user-friendly preparation and presentation of demanding control design and optimisation strategies persist. Considering the wide spectrum of methods used and techniques available for complex control problems, the MPC packages are dominant. Other methods are mostly available for process and application specific solutions.

The above review of controller design tools gave a general idea of the work that has been done so far and the future potential. In general, most tools have been aimed at control engineering experts in industry (particularly fuzzy control and neural network control) who want to focus on the controller design task without the requirement of both extensive programming skills and a substantial amount of time for low level interaction. However, there is an overall issue that has received very little attention: To support industrial standard users like area engineers that require more individual support than the industrial control engineer.

In the light of present day requirements, there is a strong economical and ecological need to replace the prevailing 'rule of thumb' approach of area engineers, with more systematic methodologies. The review was focused on the controller design methods and tools for nonlinear and multivariable processes, useable by industrial users without control expert knowledge. The ease and transparency of the reviewed controller design methods for industrial users without control expert knowledge is questionable. Table 3-6 gives a brief conclusion of the reviewed methods and tools under this focus.

For an industrially used control design tool it is beneficial to provide a relatively small collection of selected methods which are robust in application and which can be applied to a wide range of problems yielding sensible results. In this way mathematical methods can help to reduce the software complexity – complexity, which is arduous for the programmer and also for the user.

Table 3-6 Reviewed methods and tools

Controller design method or tool	Nonlinear	MIMO	Useable for non expert
SISO - PID	-	-	✓
MIMO - PID	-	✓	-
MPC	✓	✓	-
Fuzzy control	✓	✓	-
Neural network	✓	✓	-

PID-based control systems for nonlinear multivariable processes may become a practical solution for the practical engineer to solve complex control tasks, if he is efficiently supported by an appropriately refined industrial CACSD system. The following chapters will challenge control system designers from the perspective of CACSD for industrial use. A concept for an industrial CACSD procedure to solve nonlinear MIMO control design problems is derived to provide an alternative for the oversimplified identification and control design strategies often being applied in process industry and also for the complex controller design procedure discussed in this section. Additionally and specifically suitable methods for the controller design of nonlinear and multivariable processes are proposed, which have been modified for simplified use. Finally a prototype realisation is presented to validate the approach presented.

4 Proposal for a New Approach to Industrial Controller Design

The conclusion drawn from the foregoing literature review, is that an industrially applicable systematic control system design procedure for conventional PID-based controllers in nonlinear SISO and complex MIMO processes is desirable. This systematic controller design procedure must be understandable for industrial users with limited control expertise. In this chapter an Industrial Computer Aided Control System Design (*Industrial CACSD*) scheme is proposed that sets the starting point for the development of a structured approach to controller design techniques for the analysis of industrial processes, which follows the pragmatic engineering approach of 'starting simple and adding complexity only if necessary'.

In order to enlarge upon the design transparency for industrial users, a stepwise approach was designed, which reflects the user's thinking by increasing the complexity of the control system progressively. The standardised control system design procedure was designed for control systems based on advanced PID-based control schemes thus avoiding heuristic PID tuning by 'trial and error'. The approach is described in the next sections.

4.1 Introduction to the Industrial CACSD Scheme

In Chapter 2.2. it has already been noted that standard control schemes are available for some multivariable processes. However, if these standard control schemes are not accessible or become invalid, for example if the process is redesigned, a new control system design must be elaborated. Mostly the first step in practical control system design is based on the simplifying assumption that it is sufficient to split the multivariable process into independent singlevariable subprocesses by associating each process output to be controlled, to the process input with the greatest influence on it. For each of these independent SISO paths a separate PID controller is designed on the basis of rather rudimentary process information (Schumann *et al.*, 1996). Only if this approach fails due to unacceptable control performance will a deeper process analysis be undertaken. Then observed changes in process gains and time constants as well as coupling effects between the SISO subprocesses are considered in addition, for the control system design to cope with the observed effects. Over time quite complicated control schemes that could become standard for specific processes might develop.

Considering the above described industrial design path the control system design should not start with the assumption of a complex process model block, representing a highly complex nonlinear MIMO process, but a stepwise approach is preferable. Supporting the industrial user, the control scheme should start simple and add process complexity in the form of function blocks if they become necessary. This incremental approach considerably facilitates transparency for the user.

Therefore within the research group linking the University of Glamorgan and the Fachhochschule Hannover, an *Industrial CACSD* scheme has been developed (Schumann *et al.*, 1996). This *Industrial CACSD* scheme is streamlined to support the industrial user in their traditional control system design for MIMO processes and supports them in identifying the processes, generating process models and designing industrial control systems. The work conducted by the author and reported in this thesis was primarily concerned with the controller design phase of the *Industrial CACSD* scheme (Figure 4-1), although the interface between ICAI and ICAC was also part of the authors responsibility. A graphical user interface provides, for every step of the systematic control system design procedure, one default preparametrised identification and control design method. The industrial user should not be confused by different identification and design methods or the specification of too many design parameters. The aim is to enable a more efficient and reliable solution for complex industrial controller design tasks than is possible by purely manual design.

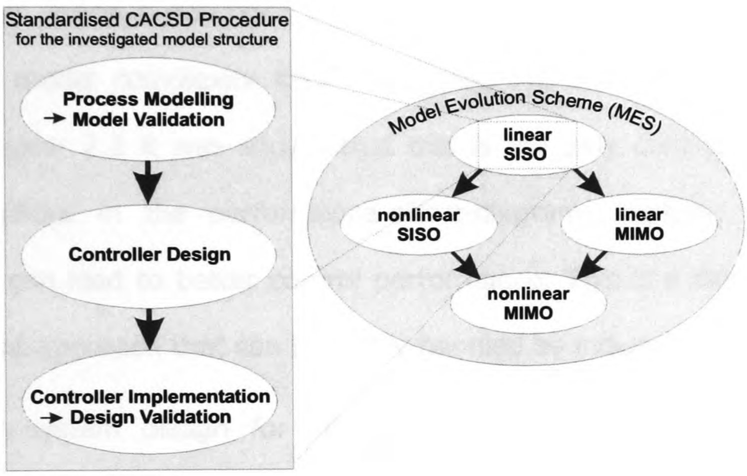


Figure 4-1 Industrial CACSD model evolution scheme and standardised CACSD procedure

The pragmatic design procedure for industrial users has been adopted in order to avoid heuristic PID-based control system design and to improve the support for process personnel. The resulting *Industrial CACSD* scheme depicted in Figure 4-1 is based mainly on two principles:

- a standardised CACSD procedure with reduced degrees of freedom with respect to process model generation and control scheme design.

- a model evolution scheme (MES) which includes four standard process model structures with complementary control system structures, yielding the simplest solution with acceptable control performance.

4.1.1 The Model Evolution Scheme (MES)

This part of the *Industrial CACSD* scheme is called model evolution scheme (MES) because it defines the complexity of the model, on which the control system design is based. In this sense it describes the control system complexity more than the actual process model complexity itself. Advanced PID control has been selected because in Chapter 2.2 it was shown that this is the only control technique that displays a gradient in the performance-effort-diagram, implying that a better process model can lead to better control performance. This is a desirable property for a subsequent approach that can be easily handled by industrial users.

In the control system design for MIMO processes, it is natural for industrial engineers to split the process into independent SISO sub-processes. For each SISO sub-process an individual PID controller is then designed, often from rough estimates of dominant time constant and gain of the investigated sub-process. Final control parameterisation is done by rules of thumb or by simple manual tuning of the controller. Only if this simple approach fails, will the controller design procedure be extended to introduce compensations for the nonlinear effects or decoupling. These "hand-tuned" rules fail in many cases and the numerical optimisation methods which are implemented in many industrial control systems display unsatisfactory control behaviour because these methods are rather time consuming and may not find the global minimum (Hensel and Peter, 1984). The independent SISO sub-process model is referred to in the model evolution scheme as the linear

SISO model. The standardised *Industrial CACSD* procedure is called for each structure in the model evolution scheme serving as the main vehicle to produce models and corresponding control structures.

If the first step within the model evolution scheme, i.e. *linear SISO*, is concluded to be unsatisfactory, the next model structure is evaluated. If nonlinearities effect the model quality significantly, then nonlinear elements will be added to the linear SISO model. This is referred to as the *nonlinear SISO* model in the evolution scheme. If the process shows linear behaviour in the working range and coupling affects the control performance, linear models representing the cross-coupling are identified in addition and built into the process model. This is referred to as the *linear MIMO* model in the model evolution scheme. Finally if the model at this MES-level fails, or if coupling and nonlinear effects are detected both as being significant, then the most complex structure, that means the nonlinear MIMO model, is applied for identification. At this stage the process model consists of input or output nonlinearities and the linear dynamic multivariable model with cross-couplings.

4.1.2 The Standardised CACSD Procedure

Another important deviation from the intuitive, heuristic approach in industry is the application of a control system design procedure, which has been standardised for the class of model structures outlined in Section 4.1.1. This facilitates the use of simple methods for identification and control system design. Naturally, a standardised and therefore simplified *Industrial CACSD* procedure – based on simple process models – cannot guarantee an optimal result which may be possible with more degrees of freedom. However, for the commissioning of industrial processes a good control system design procedure following a standard design path may be the best result achievable based on circumstances of limited expertise and with little time available. Furthermore the standardised *Industrial CACSD* procedure provides good reproducibility of results, by restricting the variety of possible solutions. Within this work, a basic form of the standardised *Industrial CACSD* procedure has been developed for each level of the MES.

The *Basic Standardised CACSD Procedure*

The so-called basic standardised CACSD procedure consists of the three main CACSD phases as outlined in Figure 4-1, Phase 1 'Process Modelling', Phase 2 'Controller Design', Phase 3 'Controller Implementation'. It is applied for each level in the MES serving as the main vehicle to produce models and corresponding control structures. This procedure is rather inflexible and specifically aimed at process personnel as described by Körner *et al.* (1996). After the process identification, the control system performance can be predicted by simulation of the complete control system with process model and controller. If the simulated control performance is satisfactory the control system can be tested on the real process. If the control performance on the real process is not satisfactory the procedure would be repeated at the next level of the MES until the final level of the model sophistication is reached. Naturally, one must consider that the control system has only been designed for those effects covered by the identified process model structure used and that the controlled process may behave differently to the simulation. This means that the agreement between simulated and actual control behaviour at the first two levels of the model evolution scheme may be quite poor.

For this reason an improved standardised CACSD procedure has been developed.

The *Improved Standardised CACSD Procedure*

The improved standardised CACSD procedure is based on the same set of model structures but it utilises an advanced identification scheme within phase 1, namely the **S**tandardised **N**onlinear **I**dentification **P**rocedure called SNIP (as proposed by Körner, 1999). The SNIP supports the process modelling for multivariable, linear, Wiener- and Hammerstein-models. Hence the control design is based on a more satisfactory process model. The SNIP Phase 1 of the standardised *Industrial CACSD* procedure generates only the most complex model. Therefore it is no longer necessary to execute all *Industrial CACSD* phases at each level of the MES successively. As the process is analysed within SNIP with respect to the MES, the controller can also be designed for the detected process model complexity directly. However, it might be helpful to undertake the control design, step by step for each level of the MES, up to the detected process model complexity, because then it is possible to check and visualise the stepwise increase in control performance in order to select the necessary control complexity. Additionally the simulation results are more trustworthy than those of the basic standardised CACSD procedure because of the improved process model quality.

4.2 Industrial Identification

The most important requirement for systematic analysis and design of control systems is the availability of comprehensive dynamic process models. The *Industrial CACSD* scheme in its current state of development comprises three CAE tools for the generation of process models (Figure 4-2).

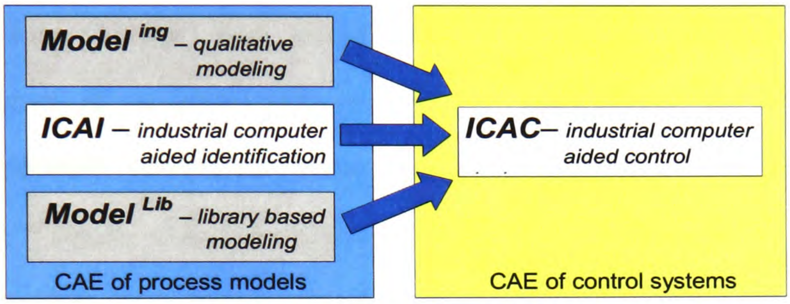


Figure 4-2 Industrial CACSD modules

- the *Model^{ing}* toolbox for the qualitative design of process models using the process knowledge of the industrial user (Schumann *et al.*, 1996). The knowledge engineering approach of Model^{ing} allows direct interaction with the area expert (i.e. the industrial user) without the need of a knowledge engineer who would normally be responsible for the translation of the unstructured knowledge of the area expert into a formal representation (or process model) (Figure 4-3). As the primary information source of Model^{ing} is the experience of the area engineer, the Model^{ing} approach is applicable to a

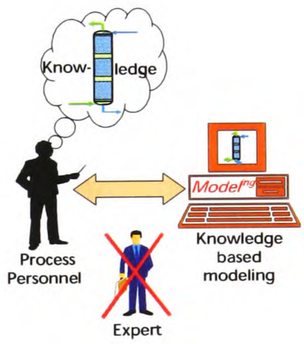


Figure 4-3 Knowledge based process Model^{ing}

process which is already in operation or even during the planning phase of a standard process provided that sufficient operation experience has already been gained from similar processes, by the area engineer (Strickrodt, 1997).

- the MATLAB™ toolbox *ICAI* (Industrial Computer Aided Identification) toolbox for industrial computer aided identification (Körner and Schumann, 1997, Körner *et al*, 2000) (Figure 4-4). ICAI generates structured linear and nonlinear single- and multivariable process models from measurement data of the process. ICAI is therefore especially useful during start-up and operation of the process where it can also serve to analyse the behaviour of the real process (Körner, 1999).

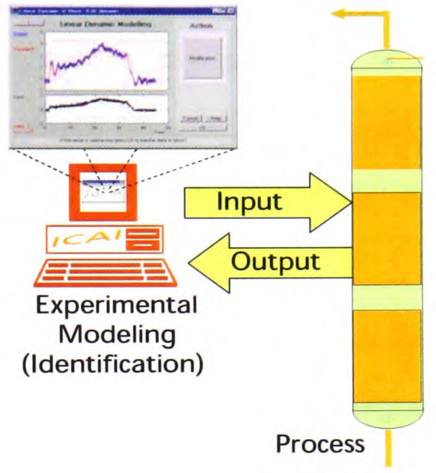


Figure 4-4 Process modelling with ICAI

- the newly specified but yet to be realised *Model^Lib^* toolbox aimed at the aggregation of process models from subsystem (component) models (Syska *et al.*, 1999). The basic idea of Model^{Lib} relies on the definition of an electronic catalogue of process components like valves, pumps, pipes, superheaters etc., including dynamic models of the

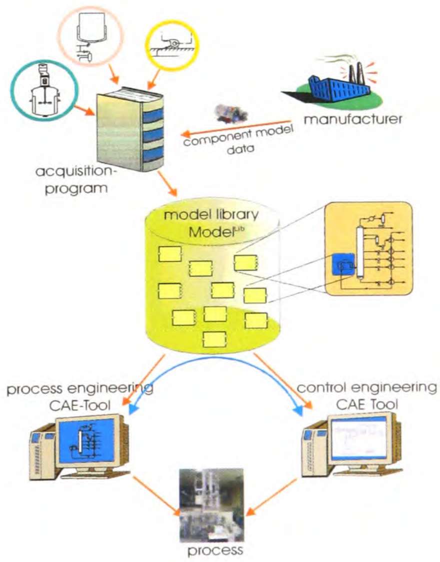


Figure 4-5 Library based modelling

components' behaviour as required for the control system design. Such an electronic catalogue would be supported by the component producers who would contribute to it by supplying models and providing all technical information required for planning and maintenance of the process control system. During the process planning, where the process is composed of such components, $Model^{Lib}$ would automatically generate dynamic models from the components submodels. $Model^{Lib}$ is currently the subject of another PhD study (Hoyer *et al.*, 2003).

From the literature review and the discussion of the practical requirements (Körner, 1999) it was shown that simple Wiener- and Hammersteinmodels are especially suited to system identification aimed at industrial control system design. The well-known Wiener and Hammerstein systems are nonlinear models that are used in many domains for their simplicity and physical meaning, where the system steady-state behaviour is completely determined by the static-nonlinearities, while the system dynamic behaviour is determined by both the static nonlinearities and the linear dynamic model components. For example, a Wiener process model consists of a linear dynamic block followed by a nonlinear static block. A Hammerstein process model is just a Wiener process model structurally reversed, that is, a nonlinear static block is followed by a linear dynamic block.

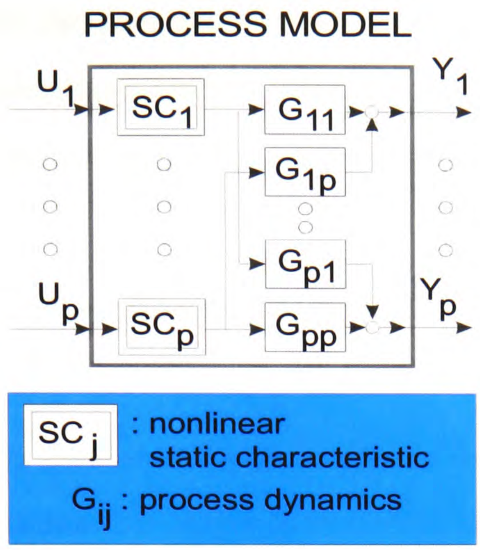


Figure 4-6 Hammerstein-model in a cross coupled multivariable structure

In this research project it is preferred that process models should be in the form produced by the MATLAB™ toolbox *ICAI*. *ICAI* generates standardised model structures composed of linear dynamic SISO blocks and, if necessary, nonlinear static SISO blocks, possibly resulting in simplified Wiener and Hammerstein models, as depicted in Figure 4-6 (for more information see Körner *et al.,* 1997). Other process models generated from other CAE tools and also general models, which can be simulated, could be preprocessed with the ICAI toolbox to get this kind of structured process model.

4.3 Requirements for an Industrial Computer Aided Controller Design Procedure

This research project is focused on computer aided controller design within the *Industrial CACSD* framework. There are several tasks to be performed with respect to a software program for industrial controller design, if the proposed *Industrial CACSD* scheme is to be applied usefully. It has been shown in chapter 2.1 that it is desirable to support the development of advanced PID control system design strategies. With the standardised CACSD procedure the process model structure should be directly reflected in the associated control system which is a combination of:

1. linear single variable PID controller blocks tuned for the linear dynamic part of the process model
2. nonlinear static blocks defined as inverse blocks of the corresponding process model nonlinearities and/or
3. in the MIMO case linear feedforward compensating blocks tuned to reduce effectively the cross-coupling effects.

The main task for the development of a software program for industrial controller design is the definition of a standardised control system design procedure, where the following needs must be taken into account.

- The control system design task must be integrated into the *Industrial CACSD* environment with an industrial graphical user interface (GUI) which guides the user carefully, especially in the case of complex MIMO controller design tasks, through the standardised controller design procedure.
- The generated process models within the *Industrial CACSD* environment should be automatically transferred to the controller design module.
- In order to make the application of control system design as simple as possible for industrial users, only few well tested control system design methods (numerical and analytical) with only few specification parameters should be provided.

- To realise this approach for industrial control system design it is not sensible to complicate the user interaction offering a variety of functions, as most existing CACSD programs do. Moreover the functionality needs to be automated and hidden so that the user has only a few but powerful functions that can be handled easily. In this sense the program must be accessible through a graphical user interface that is task-oriented rather than algorithm-oriented.
- The resulting control system, however, should consist in every case of linear PID-type controllers possibly combined with nonlinear characteristic blocks, with which an industrial user is likely to be familiar and which furthermore can be realised with available industrial process control systems (PCS).

As part of the research methodology this program was tested in student's laboratory courses to verify the proposed approach.

4.3.1 Control System Structure

The MATLAB™ toolbox *ICAI* generates four kinds of standardised process models as described in chapter 4.3. The control systems which are designed for the use within conventional process control systems are correspondingly simple and comprehensive to the *ICAI* process model (Syska, 2002) and are described for a nonlinear MIMO process model as follows:

The MIMO process model consists of nonlinear static blocks (static characteristics), main linear dynamic blocks and a linear dynamic coupling network. The complementary control system includes main linear PID controller blocks (PID), a decoupling network and nonlinear static blocks (inverse static characteristics).

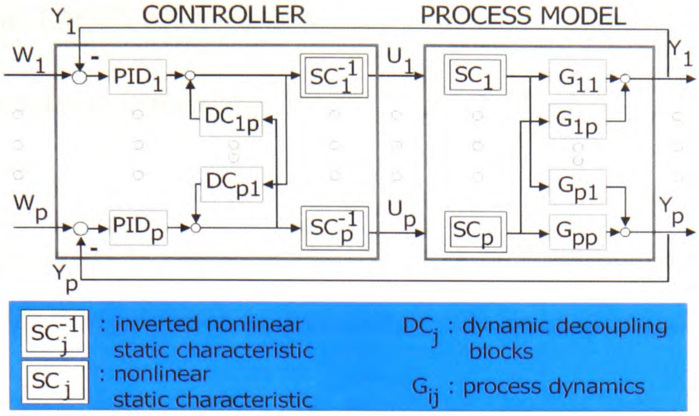


Figure 4-7 Nonlinear MIMO process model with complementary nonlinear MIMO control system

4.4 Conclusion of this Chapter

An *Industrial CACSD* scheme, consisting of a standardised CACSD procedure and a model evolution scheme, has been described. The model evolution scheme reflects the traditional way of doing control system design in a systematic way. The standardised CACSD procedure only supports a restricted model complexity that can easily be handled by inexperienced users and it provides a good reproducibility of the gained results by limitations the variety of possible solutions.

A basic and improved standardised CACSD procedure has been introduced and the advantage of the improved procedure has been discussed. However, the improved procedure necessitates a standardised control design procedure, which is introduced in the next chapters.

During practise courses in the control laboratory of the Fachhochschule Hannover students reported the experience, that the stepwise increase of control performance following the model evolution scheme is helpful to get an impression of the process behaviour and the possibilities of advanced PID control system design.

Using conventional CACSD/CACE tools students complaint about:

- o Interfacing problems concerning file format, data structure and the meaning of parameters.
- o Different user interfaces and different utilisation of the software. The user has to keep track of all the programs and underlying program philosophies.

These problems were addressed by the prototype development carried out at the University of Glamorgan and the Fachhochschule Hannover. Part of this prototype is the MATLAB™ toolbox *ICAC* outlined in Chapter 7, which utilised the standardised control system design procedure as a main prerequisite for the application of the improved standardised *Industrial CACSD* procedure.

5 Proposed Preparation of Nonlinear and Multivariable Process Models for the Industrial Standardised Controller Design Procedure

In this chapter a new simplified approach to the preparation of nonlinear and multivariable process models is introduced. The identification of multivariable process models and the design of multivariable controllers needs special process models usable for the controller design which are already described by Ackermann (1983) and Schumann (1982). This is the reason why only the process models that are used and the control design procedure (Chapter 6) are introduced, without detailing.

Frequently used process models for a time efficient multivariable process identification, as e.g. the simplified p-canonical structure (shown in Figure 5-3), are often a non minimal presentation. On the other hand for the controller design in state space a controllable and observable process model is desirable which can be handled by a minimisation procedure. The process models from the MATLAB™ toolbox *ICAI* are already reduced to this form.

After the first section presenting the components of the method, such as conversion of transfer function to a state space model, extension of the state space model and modification with nonlinear process model parts, the structural properties of transfer function representations and the properties of state space models, multivariable processes are introduced.

5.1 Components of the Method

In this section the components of the method for the approach of an industrial standardised controller design procedure for nonlinear and multivariable processes will be introduced. The introduction is presented for the SISO case, because the MIMO case would not have the same clarity.

The components of the methods include conversion of transfer functions to state space models. The modifications and extensions to the method are introduced. The components of the method presented for this approach were programmed in MATLAB™ as new functions, modified standard functions or by using MATLAB™ standard functions.

5.1.1 Transfer Function to State Space Conversion

A transfer function is an alternative way of representing the relationship between the input and output variables of a linear time-invariant system. In modern control theory and especially in the design of multivariable control systems the state space representation plays an important role. There are several ways to present the state space representation.

The state space representation is based on the description of a ordered system of n 1st order differential equations. The output variables of these differential equations are the state variables and they are denoted with $\underline{x}$. The state variable $\underline{x}$ is a vector of n elements $x_1, x_2, \dots, x_n$ (Föllinger,).

The MATLAB™ toolbox *ICAI* generates a continuous time model within the linear dynamic blocks, decribed as transfer function by

$$G(s) = \frac{y(s)}{u(s)} = \frac{\beta_0 + \beta_1 s + \beta_2 s^2 + \ldots + \beta_{n-1} s^{n-1}}{\alpha_0 + \alpha_1 s + \alpha_2 s^2 + \ldots + \alpha_{n-1} s^{n-1} + \alpha_n s^n}$$

(5-1)

Directly denominated at numerator by α_n, the transfer function is normalized to

$$G(s) = \frac{y(s)}{u(s)} = \frac{b_0 + b_1 s + b_2 s^2 + \ldots + b_{n-1} s^{n-1}}{a_0 + a_1 s + a_2 s^2 + \ldots + a_{n-1} s^{n-1} + s^n}$$

(5-2)

with $b_0 = \frac{\beta_0}{\alpha_n}, \ldots$ and $a_0 = \frac{\alpha_0}{\alpha_n}, \ldots$

The representation of the transfer function in state space will be realised with this normalised transfer function. It is helpful to describe *$G_1(s)$* using the transfer function

$$G_1(s) = \frac{1}{a_0 + a_1 s + a_2 s^2 + \ldots + s^n}$$

(5-3)

The output variable y(s) then results from Equation (5-3) in Equation (5-2)

$$y(s) = b_0 G_1 u(s) + b_1 s G_1 u(s) + \ldots + b_{n-1} s^{n-1} G_1 u(s)$$

(5-4)

By introducing the new variables x_i, where i = 1...n:

$x_1(s)=G_1(s)u(s)$

$x_2(s)=sG_1(s)u(s)$

$x_3(s)=s^2G_1(s)u(s)$

$\vdots$

$x_{n-1}(s)=s^{n-1}G_1(s)u(s)$

(5-5)

it is obvious that

$$
\begin{aligned}
&sx_1(s)=x_2(s),\\
&sx_2(s)=x_3(s),\\
&\vdots\\
&sx_{n-1}(s)=x_n(s)
\end{aligned}
\tag{5-6}
$$

and in the time domain

$$
\begin{aligned}
&\dot{x}_1 = x_2,\\
&\dot{x}_2 = x_3,\\
&\vdots\\
&\dot{x}_{n-1} = x_n.
\end{aligned}
\tag{5-7}
$$

The missing differential equation for x_n can be found using

$$x_1(s) = G_1 u = \frac{1}{a_0 + a_1 s + \ldots + s^n} u \tag{5-8}$$

This leads to

$$s^n x_1 + \ldots + a_1 s x_1 + a_0 x_1 = u$$

respectively with Equation (5-6)

$$s x_n + a_{n-1} x_n + \ldots a_1 x_2 + a_0 x_1 = u\,.$$

The transformation into the time domain results into

$$
\begin{aligned}
&\dot{x}_n + a_{n-1} x_n + \ldots + a_1 x_2 + a_0 x_1 = u,\\
&\dot{x}_n = u - a_{n-1} x_n - \ldots - a_1 x_2 - a_0 x_{1.}
\end{aligned}
\tag{5-9}
$$

Similarly, from Equation (5-4) follows directly

$$y = b_{n-1} x_n + \ldots + b_1 x_2 + b_0 x_1 \tag{5-10}$$

Controllable Canonical Form (Regulator Form)

The vector differential and the output equation of the *controllable canonical form (regulator form)* can be presented directly from Equation (5-9) to

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u \quad \text{or in short} \quad \underline{\dot{x}} = \underline{A} \cdot \underline{x} + \underline{b} \cdot u$$

(5-11)

and Equation (5-10) can be written as

$$y = \begin{bmatrix} b_0 & b_1 & \cdots & b_m \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} \quad \text{or} \quad y = \underline{c}^T \cdot \underline{x} + \underline{d} \cdot u$$

(5-12)

With

$\underline{x}$ state vector

$\underline{A}$ system matrix

$\underline{b}$ input vector

u input variable

y output variable

$\underline{c}^T$ output vector

$\underline{d}$ input-output vector

see e.g. Isermann (1981)

The idiosyncrasy of the controllable canonical form is that the matrix equations contain only ones and zeros and the coefficients of the transfer function. In matrix $\underline{A}$ there are only the coefficients of the nominator and in the vector $\underline{c}^T$ only the coefficients of the denominator of the transfer function.

Transformation of State Variables

It frequently happens that the state variables used in the original formulation of the dynamics of a system are not as convenient as another set of state variables or that for specific purposes special state space model structures are advantageous. Instead of having to reformulate the system dynamics, it is possible to transform the matrices $\underline{A}$, $\underline{b}$ and $\underline{c}^T$ of the original model to a new set of matrices $\underline{A}_t$, $\underline{b}_t$ and $\underline{c}^T_t$ of a transformed model.

For process models in state representation several realisations are possible by applying linear transformations (Friedland, 1987)

$$\underline{x}_t = \underline{T} \cdot \underline{x} \tag{5-13}$$

where $\underline{x}_t$ is the state vector in the new formulation and $\underline{x}$ is the state vector in the original formulation. It is assumed that the transformation matrix $\underline{T}$ is a nonsingular k by k matrix, such that it is possible to write

$$\underline{x} = \underline{T}^{-1} \cdot \underline{x}_t \tag{5-14}$$

Moreover, it is assumed that $\underline{T}$ is a constant matrix. (This assumption is not necessary, however, the equations derived below will require modification to include $\dot{\underline{T}}$, if $\underline{T}$ is not constant.)

Substitution of $\underline{x}$ as given by Equation (5-14) into Equation (5-11) and Equation (5-12) gives

$$\underline{T}^{-1} \cdot \dot{\underline{x}}_t = \underline{A} \cdot \underline{T}^{-1} \cdot \underline{x}_t + \underline{b} \cdot u$$

or

$$\underline{\dot{x}}_t = \underline{T} \cdot \underline{A} \cdot \underline{T}^{-1} \underline{x}_t + \underline{T} \cdot \underline{b} \cdot u \tag{5-15}$$

$$y = \underline{c}^T \cdot \underline{T}^{-1} \underline{x}_t \tag{5-16}$$

The transformed representation then satisfies:

$$\begin{aligned} \underline{\dot{x}}_t &= \underline{A}_t \cdot \underline{x}_t + \underline{b}_t \cdot u \\ y_t &= \underline{c}_t^T \cdot \underline{x}_t \end{aligned} \tag{5-17}$$

with

$$\begin{aligned} \underline{A}_t &= \underline{T} \cdot \underline{A} \cdot \underline{T}^{-1} \\ \underline{b}_t &= \underline{T} \cdot \underline{b} \end{aligned}$$

and

$$\underline{c}_t^T = \underline{c}^T \cdot \underline{T}^{-1} \tag{5-18}$$

In matrix algebra, the dynamics of the transformed system $\underline{A}_t = \underline{T}\underline{A}\underline{T}^{-1}$ is said to be similar to the dynamic matrix $\underline{A}$ of the original system. A well-known fact of matrix algebra is that similar matrices have the same characteristic polynomial.

Canonical forms of state representation are specially structured forms of $\underline{A}_t$, $\underline{b}_t$ and $\underline{c}_t^T$. All state representations can be transformed with a regular transformation matrix $\underline{T}$. The canonical forms of state representation can be referenced in many well-known control design books, e.g. Isermann (1981).

Row Companion Canonical Form

For the approach taken in this work it is necessary to transfer the controllable canonical form into the *row companion canonical form.* The representation of this canonical form is introduced in Equation (5-19) and Equation (5-20)

$$\dot{\underline{x}}_t = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_{t1} \\ \vdots \\ x_{tn-1} \\ x_{tn} \end{bmatrix} + \begin{bmatrix} \underline{c}^T \cdot \underline{A}^0 \cdot \underline{b} \\ \underline{c}^T \cdot \underline{A}^1 \cdot \underline{b} \\ \vdots \\ \underline{c}^T \cdot \underline{A}^{n-1} \cdot \underline{b} \end{bmatrix} u$$

with

$$\dot{\underline{x}}_t = \underline{A}_t \cdot x_t + \underline{b}_t \cdot u \tag{5-19}$$

$$y_t = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix} \begin{bmatrix} x_{t1} \\ \vdots \\ x_{tn-1} \\ x_{tn} \end{bmatrix}$$

with

$$y_t = \underline{c}_t \cdot \underline{x}_t \tag{5-20}$$

A_t can be directly described from Equation (5-3) and $\underline{c}^T{}_t$ is trivial. The transfer matrix $\underline{T}$ is easily found as observable matrix Q_B with

$$\underline{Q}_B = \begin{bmatrix} \underline{c}^T \cdot \underline{A}^0 \\ \underline{c}^T \cdot \underline{A}^1 \\ \underline{c}^T \cdot \underline{A}^2 \\ \vdots \\ \underline{c}^T \cdot \underline{A}^{n-1} \end{bmatrix} = T \rightarrow \underline{b}_t = T \cdot \underline{b}$$

$$\det(Q_B) \neq 0 \tag{5-21}$$

5.1.2 Extension of the state space process model

Following Hensel (1987) a critical review of advanced PID control methods led to the decision to produce a PID based control system by designing a state space controller with integral action and an approximation of the control behaviour to get the structure of an advanced PID control system.

It is well known that a state space controller produces stationary control errors for set points w≠0. To avoid this Hensel (1987) proposed the integration of an artificial I (integral)-effective process part into the state space model.

The following structure of a state space process model presentation of the row companion canonical structure form, is extended with an integral process part at the output. For this extended process model an extended state space controller system will then be designed (see Chapter 6).

The extended vector differential equation is shown in Equation (5-22) and the output equation is shown in Equation (5-23).

$$\underline{\dot{x}} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & -a_0 & -a_1 & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ b_1 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} u \tag{5-22}$$

$$y = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} \tag{5-23}$$

5.1.3 Generalization to Simple Nonlinear Process Models

All dynamical systems are nonlinear to some extent. This means that it cannot simply be assumed that they obey the principle of superposition. For example, if the separate application of input signals u_1 and u_2 to the input of a single-input-single-output nonlinear system produces output signals y_1 and y_2 respectively, then the application of the combined input ($u_1 + u_2$) may not necessarily produce the output ($y_1 + y_2$).

As an initial example, a universally encountered form of nonlinearity is the saturation effect, in which some system variable is prevented from exceeding some limiting value however large the input applied. Every real system is capable of exhibiting this behaviour, since all physical signals have an upper limit to their magnitude. Figure 5-1 illustrates an idealised input-output characteristic for such systems. Common occurrences of this behaviour occur in mechanical systems in which end stops limit movement, and in electrical and electronic systems whose output voltage levels cannot exceed their supply voltage.

In Figure 5-1, two input levels u_1 and u_2 are shown, such that u_1, u_2 and ($u_1 + u_2$) are within the range [$-u_{sat}$, u_{sat}]. For these inputs, the system will behave linearly. However, if the magnitude of any input signal to this system exceeds u_{sat} the output magnitude will be not able to exceed y_{sat}, however large an input is applied.

This gives one indication of how this type of nonlinear systems might be handled. As long as the signal levels are kept within a certain range, the system of Figure 5-1 will behave linearly. Many systems have approximately linear behaviour for suitably restricted signal ranges. This is basically the approach taken in all linear control work – restrict the signal levels, and ignore the nonlinear effects. Fortunately this works well in many cases. It is fortunate, because most of the previous techniques studied for the systematic analysis and design of control systems only work with linear system models.

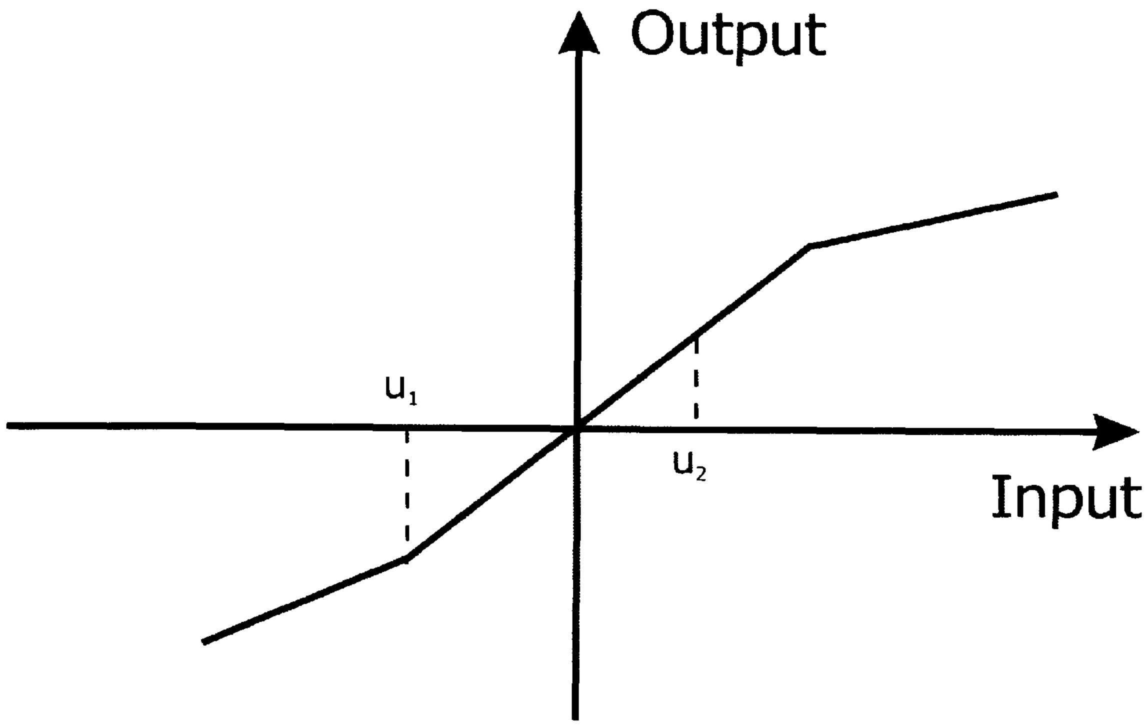


Figure 5-1 Input-output characteristic of a system element

If a system contains significant nonlinear elements, a linear model will not give a good representation of its behaviour. The model must then be modified by including nonlinear effects up to the point that the model gives a close enough approximation to the real system. Once the model has become nonlinear, all the linear control techniques begin to collapse.

Static characteristics maybe presented as look-up tables. Within the approach of this work only static characteristics which can be inverted are considered. The nonlinear effects of the process model are linearised with the inverted static characteristic such that only the linear dynamic process model part may be used for the controller design.

5.2 Representation of Multivariable Processes

As shown in Figure 5-2 the inputs u_i of multivariable processes influence all outputs y_i, resulting in mutual interactions of u_1-y_1, u_1-y_2 etc.. The internal structures of multivariable processes have a significant effect on the design of multivariable control systems. The structure in Figure 5-2 can be obtained by theoretical modelling if there is sufficient knowledge of the process. The structure of industrial processes are very different in that they cannot be described in terms of only a few standardised structures. However, the real structure can often be transformed into a canonical model structure using similarity transformations or simply block diagram conversion rules.

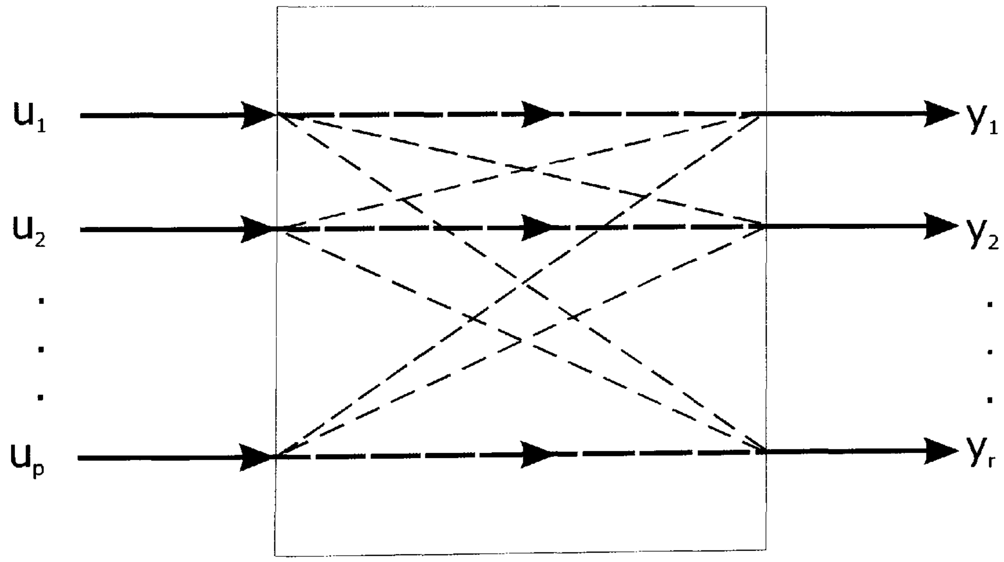


Figure 5-2 Multivariable Process

The following section considers special structures of multivariable processes based on transfer function and state space representations. These structures are also the basis for the design of multivariable control structures. Other representations will not be introduced because they are not relevant to this approach to the control system design procedure.

5.2.1 Structural Properties of a Transfer Function Representation

The MATLAB™ toolbox ICAI generates multiple-input-multiple-output (MIMO) process model as a combination of multiple-input-single-output (MISO) sub-models in p-canonical form, see Figure 5-3. Based on this process model the following continuous time transfer functions can be distinguished:

Main transfer elements, for example as 2nd order transfer function:

$$G_{11}(s) = \frac{y_1(s)}{u_1(s)} = \frac{b_1^{11}s + b_0^{11}}{a_2^{11}s^2 + a_1^{11}s + 1}$$

$$G_{22}(s) = \frac{y_2(s)}{u_2(s)} = \frac{b_1^{22}s + b_0^{22}}{a_2^{22}s^2 + a_1^{22}s + 1}$$

Coupling transfer elements, for example as 2nd order transfer function:

$$G_{12}(s) = \frac{y_1}{u_2} = \frac{b_1^{12}s + b_0^{12}}{a_2^{22}s^2 + a_1^{22}s + 1}$$

$$G_{21}(s) = \frac{y_2(s)}{u_1(s)} = \frac{b_1^{21}s + b_0^{21}}{a_2^{11}s^2 + a_1^{11}s + 1}$$

G_{11} and G_{12} are called the main transfer elements and G_{12} and G_{21} the coupling transfer elements. This example shows that there are common transfer function elements in this input/output representation. The transfer function can be summarised in a transfer matrix $\underline{G}(s)$.

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}$$

$$\underline{y}(s) = \underline{G}(s) \cdot \underline{u}(s) \tag{5-24}$$

In this example the numbers of inputs and outputs are equal, leading to a square transfer matrix. If the number of inputs and outputs are different, the transfer matric becomes rectangular. It should be noted that the transfer function elements describe only the controllable and observable part of the process. The non-controllable and non-observable process parts cannot be represented by transfer functions.

One of the most important canonical structures used to describe the multivariable process input/output behaviour are shown in Figure 5-8 (Dutton *et al.*, 1997). In the case of the p-canonical structure each input acts on each output and the summation points are at the output; p-canonical multivariable process models are described by Equation (5-24).

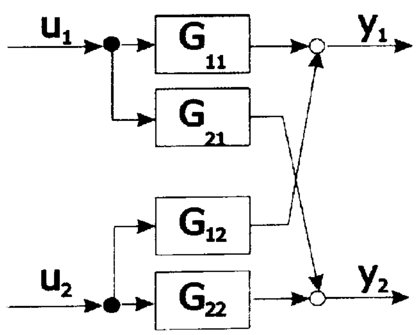


Figure 5-3 P-canonical structures of multivariable processes shown for a two-variable process

5.2.2 Structural Properties of the State Space Representation

The state space representation of multivariable systems has several advantages over the transfer matrix notation. For example, arbitrary internal structures with a minimal number of parameters and non-controllable or non-observable process parts can also be described. In modern control theory and especially in the design of multivariable control systems the state space representation plays an important role.

A linear multivariable state space model can be described by two matrix vector equations similar to those in the singlevariable case.

1. *vector differential equation*

$$\dot{\underline{x}} = \underline{A} \cdot \underline{x} + \underline{B} \cdot \underline{u} \tag{5-25}$$

2. *output equation*

$$\underline{y} = \underline{C} \cdot \underline{x} + \underline{D} \cdot \underline{u} \tag{5-26}$$

With

x	is an	(mx1)	state vector
u	is a	(px1)	input vector
y	is an	(rx1)	output vector
A	is an	(mxm)	system matrix
B	is an	(mxp)	input matrix
C	is an	(rxm)	output matrix
D	is an	(rxp)	input-output matrix

The input-output matrix D is in most cases a zero matrix. Because of this the matrix D finds no application in the further approach.

Analogously to Figure 5-3 a two-variable process with p-canonical structure is shown in its state space form in Figure 5-4. The state space representation then becomes

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \underline{A}_1 & 0 \\ 0 & \underline{A}_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{5-27}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} c_1^T & 0 \\ 0 & c_2^T \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{5-28}$$

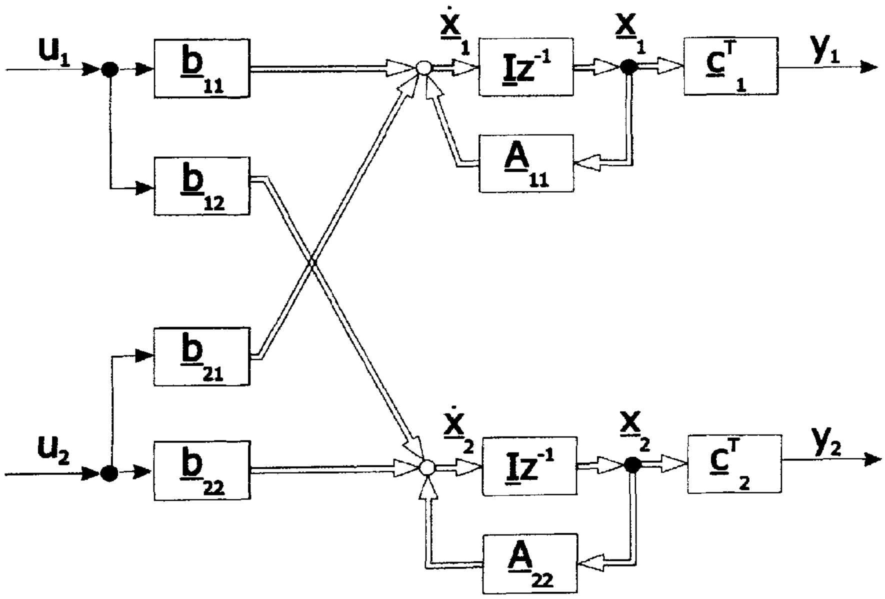


Figure 5-4 Two-variable process with a p-canonical process

The state representation with a minimal number of states is obtained and is called a minimal realisation. A minimal realisation is both controllable and observable. Nonminimal state space representations comprise therefore, either incompletely controllable and/or incompletely observable parts. From literature there are several minimisation methods which are well-known, e.g. Silverman (1971) and Nour Eldin and Heister (1980).

5.3 Conclusion of the Chapter

Within this chapter the preparation of nonlinear and multivariable process models for the industrial standardised controller design procedure was introduced. After the introduction on how multivariable processes generated from the MATLAB™ toolbox *ICAI* can be presented as transfer function matrix models and state space models, the model components for the controller design procedure are introduced. The process models available from the toolbox *ICAI* as transfer function matrices have to be converted into state space in controllable canonical form. After transformation into the row companion canonical form the state space process model is extended with an integral part on the output. In the last section the nonlinear model parts of the *ICAI* process model were introduced, generated as a static characteristic, which were linearised with an inverted static characteristic. These procedures have been developed to serve as the main methods for the control system design procedure within the prototype realisation outlined in Chapter 7.

6 Proposal for an Industrial Controller Design Procedure for Nonlinear and Multivariable Processes

In this chapter a new simplified approach for the design of controllers for nonlinear and multivariable processes in industry is introduced. This approach is aimed at avoiding the need for the detailed user knowledge of available control design tools which have been discussed previously in Chapters 2 and 3.

Conventional multivariable controllers are characterised by a given controller structure and by the choice of free parameters using optimisation criteria or tuning rules. Unlike singlevariable control systems, the structure of a multivariable controller consists not only of the order of the different control algorithms but also of the mutual arrangement of coupling elements, as considered for the process models in Chapter 5. Corresponding to the main and coupling transfer elements of multivariable processes, one may distinguish between main and coupling controllers (cross controllers). The main controllers R_{ii} (as depicted in Figure 6-1) are directly dedicated to the main elements G_{ii} of the process and serve to control the variable y_i close to the reference variable w_i. The coupling controllers R_{ij} couple the signal loops on the controller side. They can be designed to decouple the process model completely or particularly, or they can be designed to reinforce the coupling. This depends on the process, the prevailing disturbance, command signals and on the requirements placed on the control performance.

This chapter contains an advanced method for the design of multivariable processes based on the process models which would be prepared as in Chapter 5. After showing the general control system structure, the optimisation method using the state space approach with the matrix Riccati equation is introduced and in section 6.3 the approximation of the optimal state space controller to an advanced PID-controller based control system is described.

6.1 Controller System Structure

The best control performance is possibly in multivariable systems if not only the control parameters, but also the control system structure is optimised for the multivariable process. Equal to various multivariable process structures, there is also a variety of possible control system structures. Figure 6-1 shows the selected control system structure. If the decoupling controllers are omitted, a simplified control system structure can be developed with only independent main controllers, this structure can be used if the process coupling effects are weak.

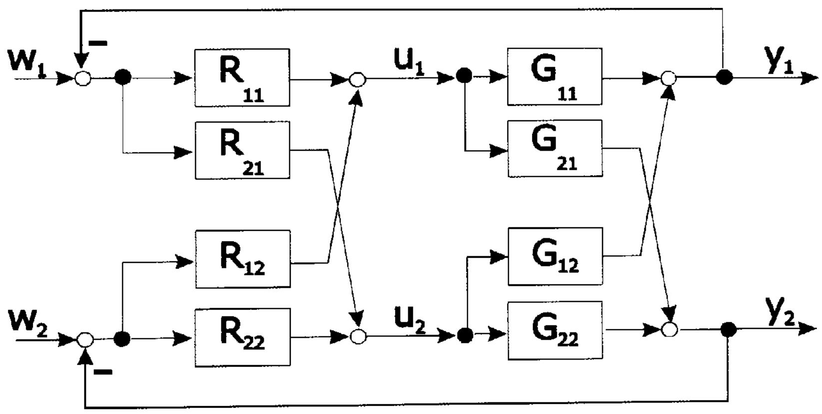


Figure 6-1 Control structure of two-variable p-canonical process model with decoupling controllers parallel to main controllers

If the coupled control system has a poor behaviour or if the process requires decoupled behaviour, decoupling controllers can be designed in addition to the main controllers.

6.2 Optimisation of the Control System

The relative quality of a control system design can be determined by a cost function that provides a figure of merit as a single number. Several typical cost functions J(u) for singlevariable case can be seen in Table 6-1 (Rowland, 1986). The first two criteria are based on minimising integrals of the squared control deviation *e* or absolute value of the control deviation. Time weights are incorporated into the next two criteria to emphasise a reduction of control deviations occurring later in time response. The next three cost functions focus on minimising energy, process input or time. The last entry in Table 6-1 is a quadratic cost function composed of the integral of a weighted combination of the squares of state variables and the process inputs. The cost function indicates how well the system performance between an initial time t_0 and a final time t_f. The design goal is to minimise the cost function.

The selection of a cost function is based on which aspects of process behaviour in a particular application are deemed most important to the control designer. If the output time response is required to approach its final value rapidly, then deviations in the state variables should be weighted heavily in the cost function.

Table 6-1 Typical cost functions

Criterion	Mathematical Expression
Integral squared error (ISE)	$J(u) = \int_{t_0}^{t_f} e^2 dt$
Integral absolute error (IAE)	$J(u) = \int_{t_0}^{t_f} \lvert e \rvert dt$
Integral time squared error (ITSE)	$J(u) = \int_{t_0}^{t_f} (t - t_0) e^2 dt$
Integral time absolute error (ITAE)	$J(u) = \int_{t_0}^{t_f} (t - t_0) \lvert e \rvert dt$
Minimum energy	$J(u) = \int_{t_0}^{t_f} u^2 dt$
Quadratic in state and control	$J(u) = \int_{t_0}^{t_f} (\underline{x}^T \underline{Q} \underline{x} + ru^2) dt$

The quadratic cost function has served as the main method for the control system parameter optimisation. This method is described in the next section.

6.2.1 Linear Quadratic Optimal Control

Consider the process model represented using the familiar state equation (Equation (5-11) and Equation (5-12)) and repeated here

$$\dot{\underline{x}} = \underline{A} \cdot \underline{x} + \underline{B} \cdot u$$
$$y = \underline{C} \cdot \underline{x}$$

The linear quadratic cost function is defined (Borrie, 1986) as

$$J(u) = \int_{t_0}^{t_1} (\underline{x}^T \cdot q \cdot \underline{x} + ru^2) dt \tag{6-1}$$

where q is a positive scalar
r is a positive scalar

$J(u)$ describes the cost of the process, $t_0 \le t \le t_1$, and comprises a positive function of the state and a positive function of the process input.

The design problem is to find an input u which drives the state $\underline{x}$ from $\underline{x}(t_0)$ to $\underline{x}(t_1)$ with minimum $J(u)$. Without loss of generality, since the states can be redefined appropriately, $\underline{x}(t_1)$ can be set at 0. If, in addition, $t_1=\infty$, the solution of the problem, which is determined by the calculus of variations, is the optimal control law (Borrie, 1986)

$$u = -\underline{K} \cdot \underline{x} \tag{6-2}$$

where

$$\underline{K} = \underline{R}^{-1} \cdot \underline{B}^T \cdot \underline{P} \tag{6-3}$$

and P is the solution of the matrix Riccati equation (Sinha, 1984)

$$-\underline{\dot{P}}(t) = \underline{P}(t) \cdot \underline{A}(t) + \underline{A}^T(t) \cdot \underline{P}(t) - \underline{P}(t) \cdot \underline{B}(t) \cdot \underline{R}^{-1}(t) \cdot \underline{B}^T(t) \cdot \underline{P}(t) + Q(t) \tag{6-4}$$

For the time-invariant case, Equation (6-4), with $t \to \infty$, becomes

$$\underline{Q} + \underline{A}^T \cdot \underline{P} + \underline{P} \cdot \underline{A} - \underline{P} \cdot \underline{B} \cdot \underline{R}^{-1} \cdot \underline{B}^T \cdot \underline{P} = 0 \tag{6-5}$$

Examples for the resulting vectors and matrices of this optimisation are shown in the sections 8.2 and 8.3 for the application example with a laboratory air-conditioning plant.

The matrix Riccati equation (Equation (6-4)) may well be the most famous control cost function in the literature of modern control theory. The benefit of the matrix Riccati equation is that the designed control system is in every case the optimal one, depending on the performance criteria. This makes the optimisation of complex MIMO control systems straightforward and the global minimum can be found. The block diagram of the optimal controlled system is illustrated in Figure 6-2.

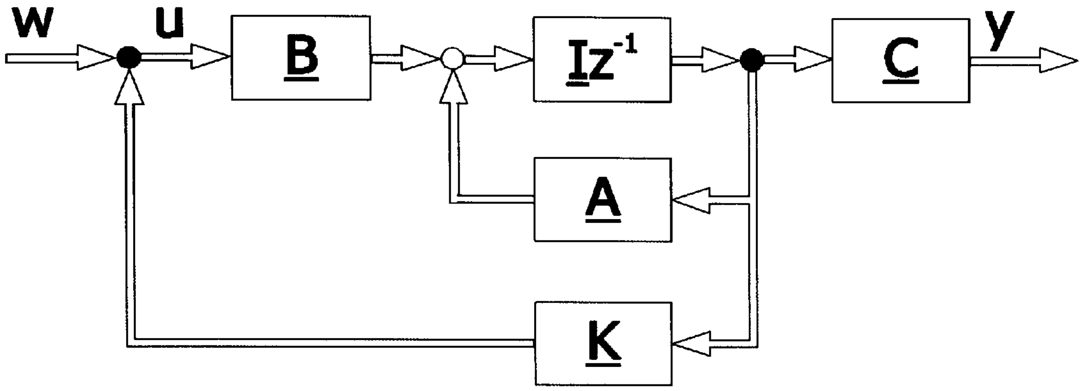


Figure 6-2 Block diagram of the optimal linear control system

6.2.2 Controller Optimisation for the Extended State Space Control System

The above section shows the general view of an optimally controlled state space system. Within the approach of this work the extended state space process model (Chapter 5.2.2), with an additional integral part at the output has to be optimally controlled. The additional integral part at the output sets an artificial additional state and the controller will also be extended by an additional controller part. The equation for the control system of the extended state space system for a SISO process is described in Equation (6-6) and the block diagram for the control system of the extended state space system for a SISO process is shown in

Figure 6-3.

$$
\begin{bmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & -a_0 & -a_1 & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ b_1 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} \begin{bmatrix} K_0 \\ K_1 \\ \vdots \\ K_{n-1} \\ K_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} y \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} w \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}
$$

$$
y = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} \tag{6-6}
$$

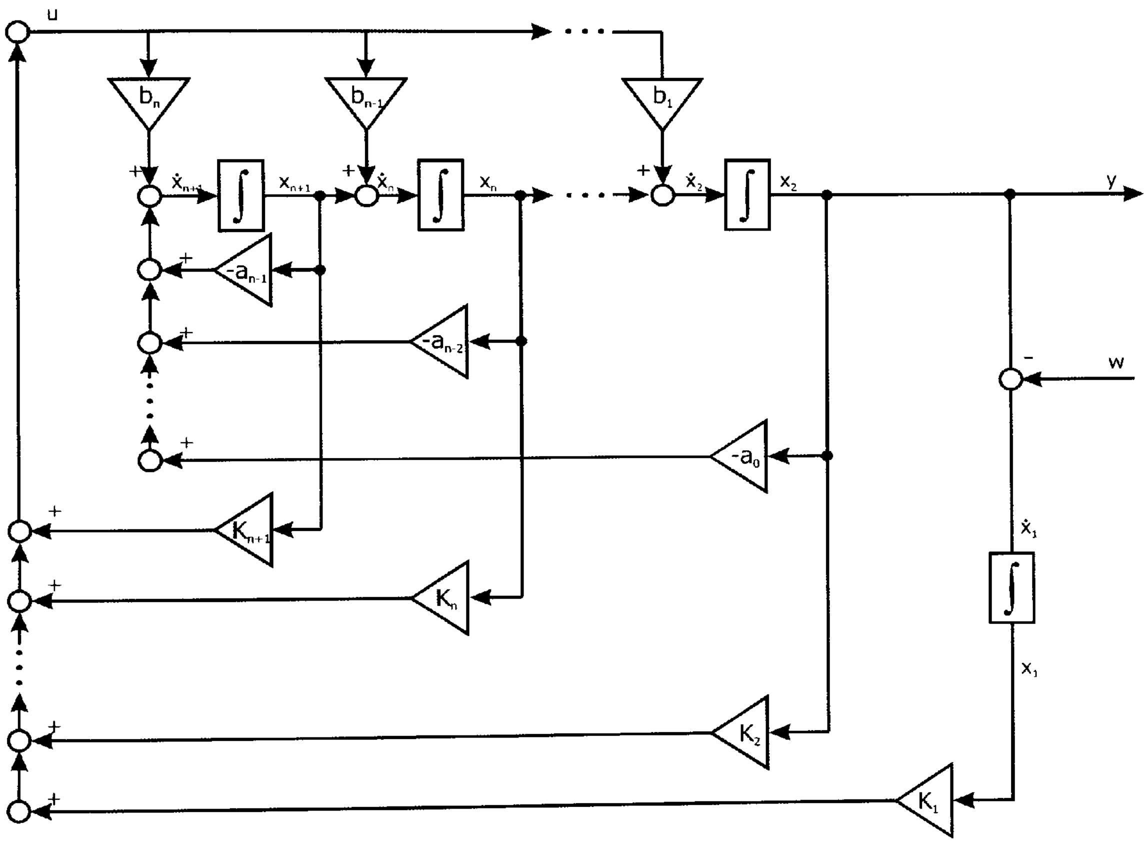


Figure 6-3 Block diagram for the control system of the extended state space system for a SISO process

For a MIMO process the equation and the block diagram should be built in the same way as is shown above for the SISO process, for each sub-process part.

6.3 Approximation of Optimal Control System by a PID-based Control System

This approach is directed at the industrial users who are not familiar with state space control systems. Within conventional process control systems (PCS) most of the control systems are realised with PID controllers. The industrial user of the methodology of this work is likely to be familiar with the use of PID controllers (Dutton,1997). Because of this the resulting control system should consist in every case of a linear PID-based control system. In this section the approximation of the control behaviour from the process model controlled by the designed optimal state space controller system using a PID-based control system is introduced. The control results which are attainable with the PID control loop depends on the approximation, which will become worse if the orders of the state space control system and the PID control system are very different.

The starting point is the extended state space presentation with an integrator at the output, Equation (5-27) and Equation (5-28), with the designed optimal state space control system, Equation (6-6). This extended state space process and control system (Figure 6-3) will be mirrored with a transfer function process and a PID control system, as shown in Figure 6-4.

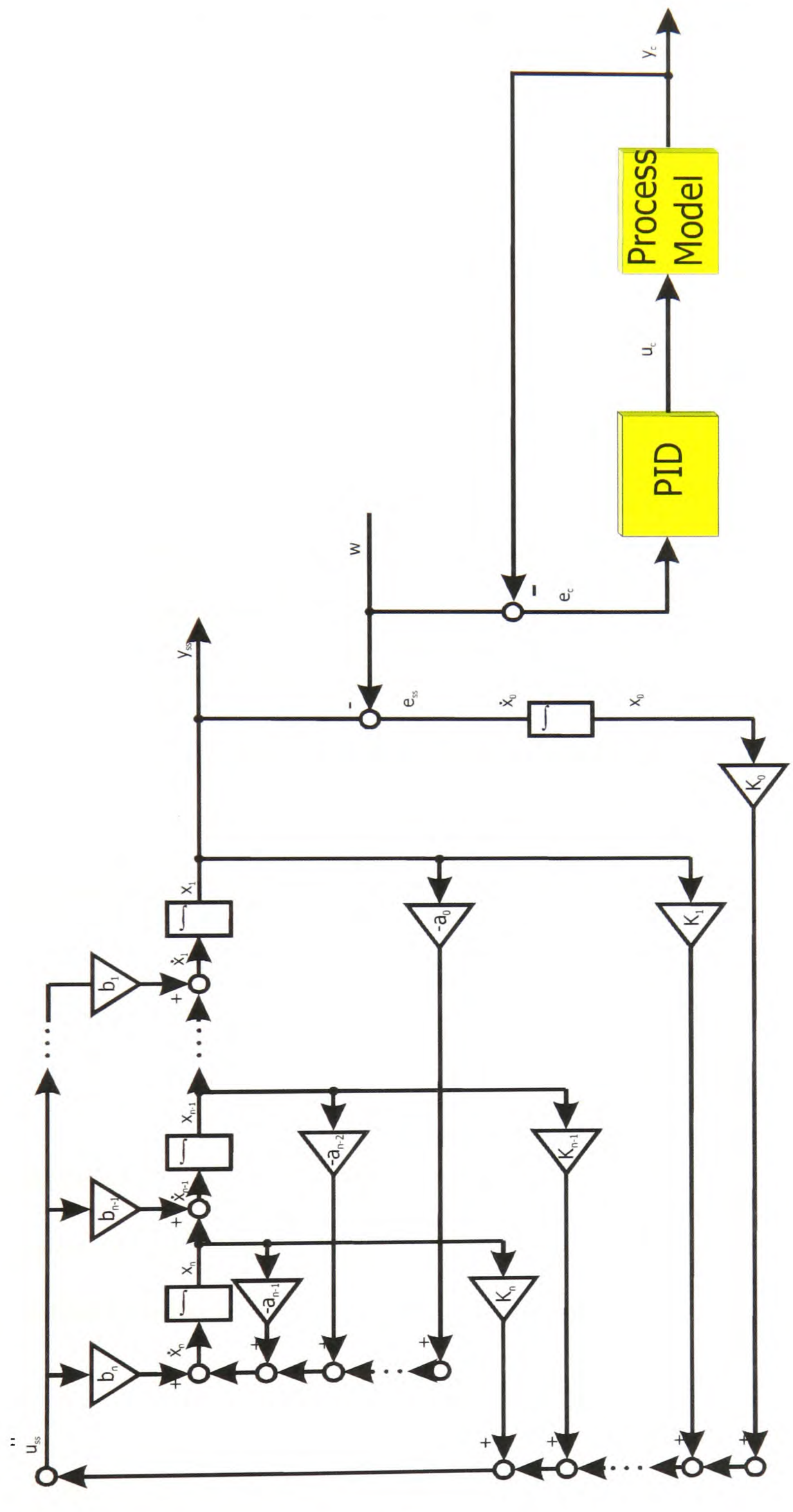


Figure 6-4 Extended state space control loop with the approximation of the PID controller parameter

The well-known PID controller can be described with the transfer function (Dutton, 1997)

$$G_C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + sK_d \tag{6-7}$$

where

K_p = proportional gain, selected for adequate rise time

K_i = integral gain (units of gain per second), selected for steady-state accuracy without making performance unacceptable poor

K_d = derivative gain (units of gain x seconds), selected to overcome excessive oscillation or too long settling time

The PID control loop should approximate the optimal control behaviour of the state space control system. The approximation and optimisation of the PID controlled process is done using the cost function

$$J(u) = E + U \tag{6-8}$$

where

E = minimisation of the difference between the minimised control deviation of the state space control system and the minimised control deviation of the PID control system, with

$$E = \int_{t_0}^{t_f} (e_{ss} - e_c)^2 \, dt \tag{6-9}$$

where

e_{ss} = control deviation of the state space control loop

e_c = control deviation of PID control loop

U = summed difference between the process input of the state space control system and the process input of the PID control system, with

$$U = \beta \int_{t_o}^{t_f} (u_{ss} - u_c)^2 \, dt \tag{6-10}$$

where

β = weighting factor (not fixed, should be bigger than 1)

u_{ss} = process input of the state space control loop

u_c = process input of the PID control loop

The PID parameters are approximated by numerical optimisation such that the control deviation and the process input of the PID control loop resembles the control deviation and the process input of the state space control system. The resulting PID parameters of this approximation are shown for the application example in the sections 8.2 and 8.3.

6.4 Conclusion of this Chapter

A standardised control system design procedure has been developed for a PID-based controller structure. The control system structure will be comprehensive to the structure of the process model. The pragmatic procedure offers the industrial user a transparent method for the control system design. The optimisation of the extended state space control system is done using the matrix Riccati equation with the quadratic optimal control cost function. The optimal control behaviour of the state space control loop is approximated by a PID-based control system. The PID parameters are approximated by numerical optimisation such that the control deviation and the process input of the PID control loop assemble the control deviation and the process input of the state space control system.

7 Prototype Realisation

Modern controller design methods are rarely used in process industry for the reasons discussed in the Chapter 3. A concept for *Industrial CACSD* has been developed (Chapter 4) and a standardised industrial controller design procedure has been proposed to promote the use of controller design methods in process industry. For the realisation of the standardised industrial controller design procedure, the preparation of the process models used and the modification of the nonlinear process model elements have been introduced (Chapter 5) and specifically methods suitable for the controller design have been selected and improved for simplified application (Chapter 6).

All this does not say anything about the applicability of the industrial standardised controller design procedure. Therefore a software prototype has been developed that in principle makes these procedures and methods accessible. The software also provides a platform to discuss the methods and strategies implemented. During the development of this prototype and, its testing, the understanding of practical problems increased considerably and many valuable ideas were developed.

The following section explains the main ideas of this prototype development. At first some general considerations for the prototype development are discussed. Then the prototype realisation is described and technical details for the practical realisation are summarised.

7.1 General Considerations for the Prototype Development

The software prototype developed should be a part of the introduced *Industrial CACSD* scheme (Chapter 4). During the development of the MATLAB™ toolbox *ICAI*

(Körner, 1999) the general framework of the *Industrial CACSD* scheme has been defined which are also applied to the new MATLAB toolbox *ICAC*, including the selection of a suitable software platform for the prototype development, an appropriate requirements analysis and the definition of some basic project guidelines, while every step within the prototype development is influenced by the customer's needs. In the following, the general considerations for *ICAI* and *ICAC* will be shortly summarized (for detailed information please refer Körner (1999)).

7.1.1 Decision for the Software Base

MATLAB™ (The MathWork 1993) has been selected as an appropriate tool to perform the prototype development. This environment entails means to implement , try and test the schemes and methods elaborated in this work. One of the main considerations for the choice were that MATLAB™ has become a quasi-standard in CACSD (Chapter 2) with the block oriented simulation environment SIMULINK and the possibility to program proprietary graphical user interfaces (GUI).

7.1.2 Main Design Principles

Some basic design principles for industrial CACSD tools have already been formulated in Chapter 2. These statements had to be refined for software development for control system design.

1. It must be tailored to industrial needs, such that even inexperienced users can intuitively use and understand the controller design routine. This means:
   - The user must be guided through standard design paths (like *the industrial control system design procedure*).
   - The use of advanced methods must be simplified utilising sensible defaults.

- Even nonlinear and multivariable controller design should be supported in a simple, transparent, reliable and reproducible procedure.
- By default, all the results should be presented in time domain through easily understandable graphs, whenever possible.

2. A primary requirement is the integration of the controller design task into a block oriented simulation environment, because block oriented simulation environments are frequently used to investigate industrial processes. Thus the user does not have to switch between different programs, when utilising an integrated solution.

A prerequisite for the realisation of these design principles is the proper design of a graphical user interface (GUI) that provides a guide through the whole controller design procedure.

7.1.3 GUI Assistance of this Project

Several ways have been tried to assist the user interacting with software programs. The most successful approach is the use of GUIs, which have been established as very helpful means to guide the user through a design procedure and to provide a context dependent assistance whenever necessary.

Principles for the GUI Design

The three main ideas of building GUIs are (sorted by importance):

1) Simplicity (guidance)
2) Consistency (handling)
3) Readability (form, colour)

4) Reliability (robustness, speed)

5) Wording (Understanding)

Naturally this is only a section of the most important aspects that have been considered while doing the GUI design. For more details please refer to Percoco and Sarti (1996) or Gram (1998).

The GUI Colour Concept

The considerations for the colour concept for *ICAC* are similar to these for *ICAI* (Körner (1999)). Many useful colour concepts have been realised. Charwat (1996) investigated aspects and remarked that the aesthetic solution by an individual is not necessarily an appropriate functional one: 'Better functional than beautiful'.

Those colours outlined in Table 7-1 have been defined for the design of the *ICAC* GUI.

Table 7-1 Colour conventions

Object	Colour	Remark
background		
figure (main window)	light grey	homogenous background without pattern or picture with similar properties as green but allowing a corporate design of the toolbox
main task	bright grey	beneath important interactive elements to attract attention, not framed
general tasks	dark grey	beneath less important elements, not framed
action field	white	to attract the user's attention to this field
curves		
input (w)	black (interrupted line)	black for the first input, if more inputs are depicted shades of grey are used
simulated output (y_{ss})	red	standard, red for the first output, if more outputs are depicted shades of orange are used
simulated output (y_c)	green	good contrast with red to recognise degree of fit; green for the first output, if more outputs are depicted shades of blue are used
blocks		
empty control block	blue	attracts attention (must still be designed)
controller block	black	looks similar to standard blocks

The User Profile

Consequently, it is necessary to base the software development on the user needs. In order to explain the general way of the *Industrial CACSD* scheme, the user-profile concept has been elaborated to specify the necessary functionalities for the *ICAC* toolbox.

The user-profile defines the degrees of freedom available for three different user levels which are already implemented within the *ICAI* toolbox:

- The *process personnel* will get a standardized path through the CACSD procedure using pre-parameterised default methods with almost no degrees of freedom.
- The *area engineer* can at least access alternative controller design methods.
- The *control expert* will have access to all design methods and parameters available.

Naturally, such standardisation cannot meet the needs of each individual user. However, it allows the design of a transparent software concept and clearly indicates that the controller design support for process personnel must be excessively automated, while control experts wish more freedom.

The *ICAC* prototype development was focussed on the process personnel requirements. The user-profiles implemented have been described in more detail in Chapter 7.2.2.

7.2 MATLAB™ Toolbox Description

The MATLAB™ toolbox *ICAC* has been designed with design principles outlined in the foregoing sections with similar functionalities and appearance to the MATLAB™ toolbox *ICAI*. In the following the specific *ICAC* functionality is described in more detail.

7.2.1 The *ICAC* (Industrial Computer Aided Control) Blockset

To allow the user modelling, identification and controller design including simulation in one single graphical environment the *ICAC* (Industrial Computer Aided Control) toolbox has been realised as a SIMULINK blockset (Figure 7-1).

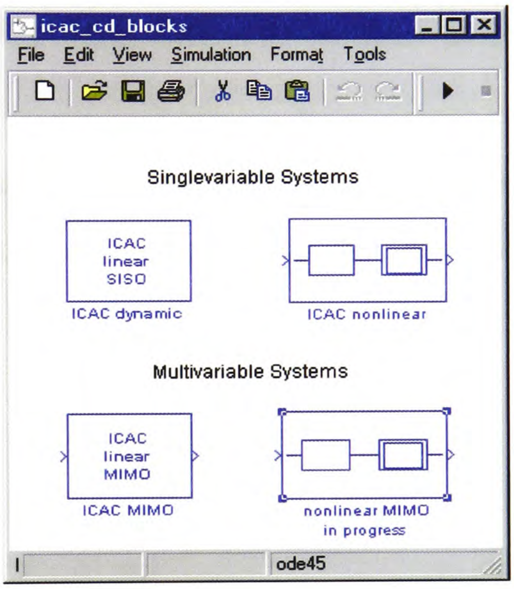


Figure 7-1 ICAC CD blocks

The *ICAC* CD (control design) blocks represent an addition to SIMULINK™ blocks and the *ICAI* ID (identification) blocks. The handling of a CD block is similar to that of other blocks. Only its colour (blue) is different to the other blocks and changes during the controller design procedure. This allows easy recognition of the status of the CD block especially within a process composed from several SIMULINK™ blocks. A double click on the control design block starts a guided tour through the control system design task utilising a graphical user interface and requesting only a minimum of information from the user.

The *ICAC* CD blockset comprises the following blocks, depicted in Figure 7-1:

SISO blocks

- *ICAC linear SISO CD* block designs a linear PID controller block
- *ICAC nonlinear SISO CD* block consists of a nonlinear static block combined with a linear dynamic PID controller block.

MIMO blocks

- *ICAC linear MIMO CD* block designs a complementary control system structure to the (structured *ICAI*) process model with main linear PID controller blocks and dynamic decoupling controller blocks.
- *ICAC nonlinear MIMO CD* block consists of nonlinear static blocks combined with designed linear dynamic PID controller blocks and dynamic decoupling controller blocks.

7.2.2 *ICAC* Project

The project settings for the *ICAC* project sets the frame for the *ICAC* controller design task. The project settings (Figure 7-2) must be specified for each *ICAC* CD block. The project settings require information about the user profile and process type.

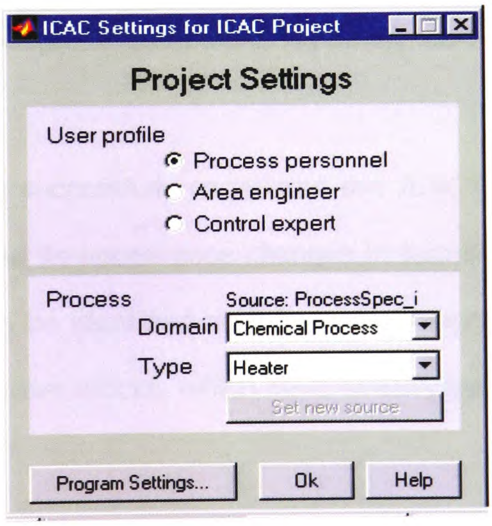


Figure 7-2 ICAC project

The user-profile includes three user levels (plant personnel - area engineer - control expert) which provide different degrees of freedom. The *plant personnel* user level restricts the degrees of freedom for users with limited control expertise such that complicated methods are hidden, and the presentation of the simulation results is only in time domain. A the *area engineer* level, additional degrees of freedom are available for the user (giving a free choice of the *ICAC* CD blocks, parameterisation of the design methods). The *control expert* level allows access to the complete *ICAC* functionality including the unrestricted but integrated use of other control tuning and optimisation toolboxes and more sophisticated presentations of the controller design results. By specifying the process domain and type, the user can add some general information about the process for which the control system is to be designed.

7.2.3 Handling of *ICAC* CD blocks

ICAC CD blocks represent just another type of SIMULINK block. The use of *ICAC* CD blocks can be compared to the use of standard SIMULINK blocks. The differences being:

- The guided control system design tour through different *ICAC* windows, started with a double click.
- When the control system design has been successfully completed the *ICAC* CD block becomes an *ICAC* controller block and its appearance changes in terms of colour and label. Unfinished CD blocks can be identified quickly, even in bigger projects and distinguished from *ICAC* controller blocks, which have already been completed.

The handling of the *ICAC* CD blocks should be simple and transparent. The utilisation of *ICAC* CD blocks is described in the following:

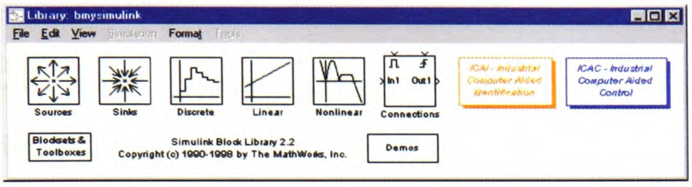
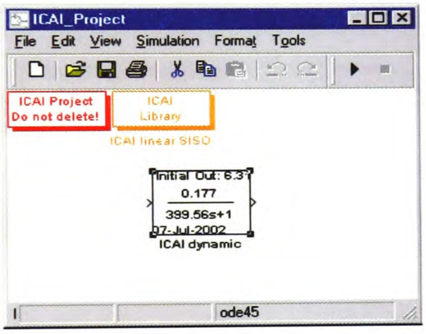
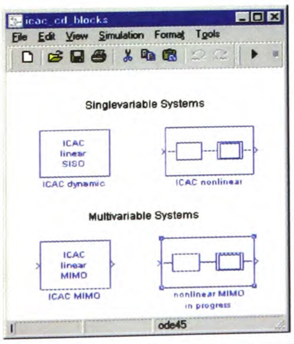
1. The *ICAC* CD block is placed into the SIMULINK block diagram as a representation of the control system. The *ICAC* CD block can already be connected to other (sub-)blocks. It can also be connected with the process model blocks in order to simulate the control system when the control system is designed.

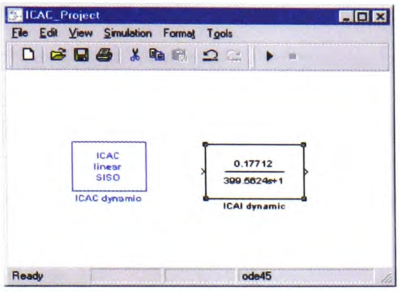
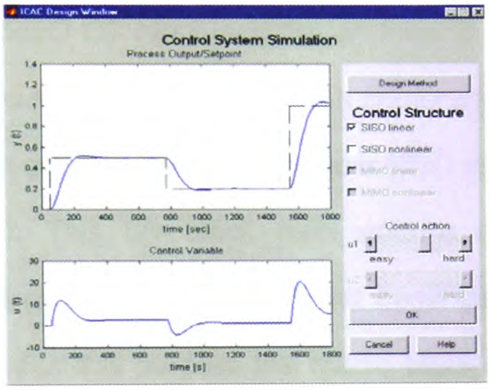
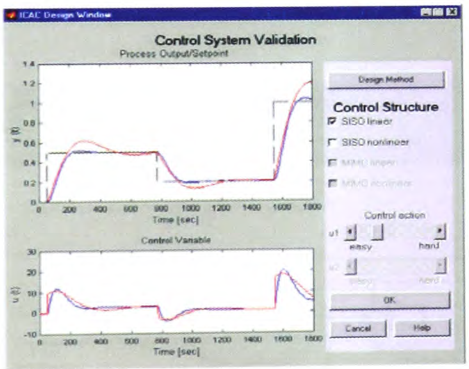
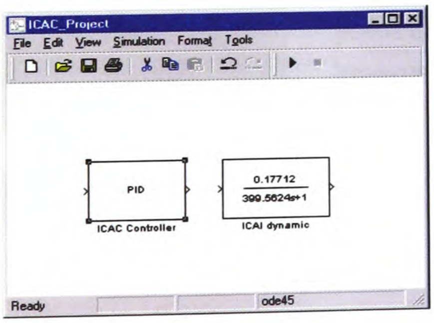
2. The prerequisite for the use of the *ICAC* CD blocks is the availability of structured process models as produced by the MATLAB™ toolbox *ICAI*. The *ICAC* toolbox inherits all important data of the *ICAI* process model, e.g. the type of process model – SISO/MIMO, linear/nonlinear – , user profile, and proposes a complementary *ICAC* CD block. Only a few relevant control system design methods are offered with extensive pre-parameterisation, such that the user has to deal with only few transparent design decisions.

3. After completion of the controller design procedure the control system block can be used for simulation or control system design within SIMULINK like any other standard SIMULINK block.

The steps of the control system design task of a linear dynamic controller using *ICAC*s linear dynamic SISO CD block are shown in Table 7-2

Table 7-2 Example for a guided tour for control system design (the linear SISO CD block)

***ICAC* GUI**	**Description**
ICACSD icon 	A double-click on the *ICACSD*-icon from the WINDOWS desktop starts
 SIMULINK with *ICAI* and *ICAC*	SIMULINK extended with the *ICACSD* tools *ICAI* and *ICAC*. All SIMULINK functions are available plus *ICAI* and *ICAC*.
ICAI Model 	*ICAI* allows modelling of a process model within *ICAI* or opens a *ICAI* project window with an *ICAI* process model (here *ICAI* linear SISO).
ICAC project 	*ICAC* starts with the project settings inherited from *ICAI* or the *ICAC* project has to be specified.
ICAC CD block library 	Then a new block diagram opens with the *ICAI* process model and the *ICAC* block library.

ICAC GUI		Description
ICAC block diagram with *ICAC* CD block and *ICAI* process model		The linear dynamic SISO CD block is copied into the new block diagram of the *ICAC*_Project.
ICAC design window		After activating the linear dynamic SISO CD block automatically an initial design result is calculated. The control performance can be modified by using the slider for stronger or weaker control action.
ICAC design window		After accepting the control performance the comparison of the optimal control system and the approximated PID controller is presented.
ICAC PID controller		After a confirmation the CD block is transformed into the *ICAC* PID controller ready for simulation.

7.2.4 *ICAC* CD Block Descriptions

The control systems, which are designed with the MATLAB™ *ICAC* toolbox for use within conventional process control systems, are correspondingly simple and complementary to the *ICAI* process models. The functionalities of the *ICAC* CD blocks are described to clarify the principal of the *ICAC* prototype as follows.

Nonlinear SISO CD Block

The *nonlinear SISO CD block* includes an inverse characteristic and linear dynamic PID controller to support the design of a comprehensive control system for Wiener- or Hammerstein process models (nonlinear SISO as identified with *ICAI* as described in section 4.2). This CD block represents a macro block that organises the successive control system design for the nonlinear and linear dynamic part of the process model. For this purpose the macro block utilises a static *ICAC* block and a linear PID controller block (Figure 7-3).

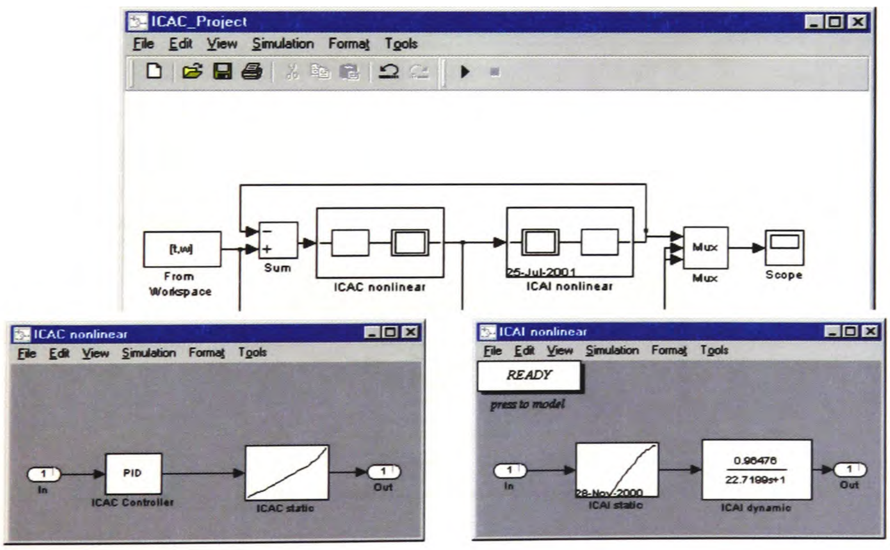


Figure 7-3 Nonlinear process model with complementary control system

Therefore the standardised control system design procedure is to design the PID controller for the linear process model part and to invert the static characteristic of the process model to produce a compensating nonlinear characteristic as part of the controller.

Linear MIMO CD Block

The *linear dynamic MIMO CD block* supports the design of a comprehensive control system for the linear dynamic MIMO process models and is composed from MISO (multiple input single output) sub-control systems as shown in Figure 7-4.

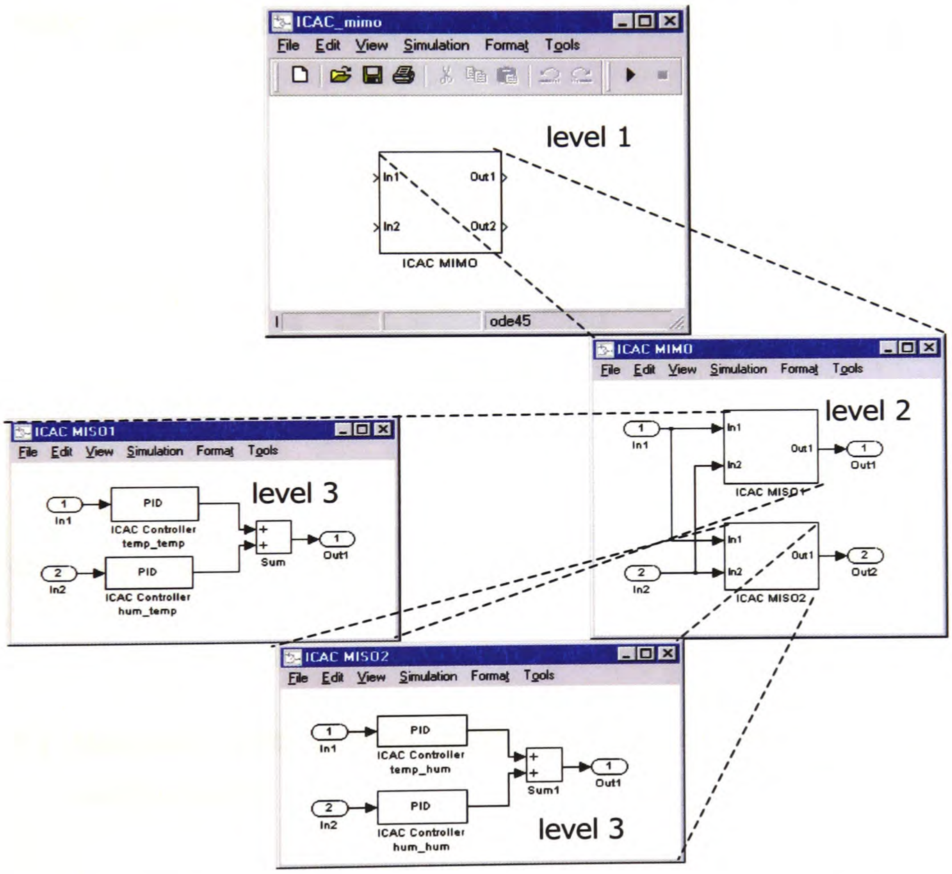


Figure 7-4 SIMULINK representation of the control system composed from MISO control systems

These MISO control systems are composed from linear dynamic PID main controllers and PID decoupling blocks. They follow a similar procedure as the linear

SISO CD block but can handle in addition an arbitrary number of input and output signals (in the prototype realisation restricted with two inputs and two outputs). The functionality is similar to that of the linear SISO CD block.

Nonlinear MIMO CD Block

The *nonlinear MIMO CD block* is a combination of the *linear dynamic MIMO CD* block with *nonlinear SISO CD* blocks and supports the design of a control system complementary to the nonlinear MIMO process model. It includes all control system blocks realisable with *ICAC* (linear dynamic PID controllers, decoupling blocks and inverse static characteristic) as it is presented in principle in Figure 7-5.

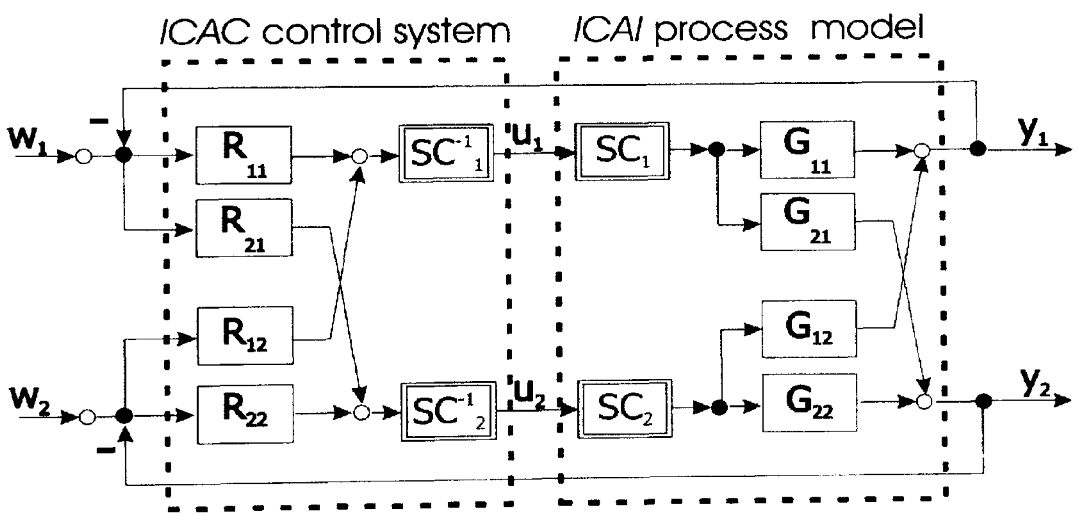


Figure 7-5 Nonlinear MIMO process model with complementary nonlinear MIMO control system

It follows a similar procedure to the nonlinear SISO CD block but it can handle in addition an arbitrary number of input and output signals. During the development of this approach, the author gained experience of systems with up to four inputs four outputs and it was noted that the problems associated with the mathematical solution of such systems were encountered. It is not possible for the author to generalise beyond four inputs and output, with respect to convergence of the algorithm, on a solution and in reasonable execution time.

7.3 Gateway to the Industrial Environment

A gateway feature for industrial process control systems (PCS) is the OPC™ (OLE for Process Control) interface. OPC™ is designed to allow client applications to access plant floor data in a simple and consistent manner. An OPC client was programmed to provide online access to PCS data – to allow the efficient reading and writing of data between a client application and the process control device.

With the online data access of the OPC™ client, test signals can be written to the PCS and process data can be read out from the plant floor. The control system designed with *ICAC* should be directly implemented in the PCS. In the current state the direct implementation of the designed control system is not allowed due to safety considerations for the process control systems. The current state of the *Industrial CACSD* scheme is presented in Figure 7-6.

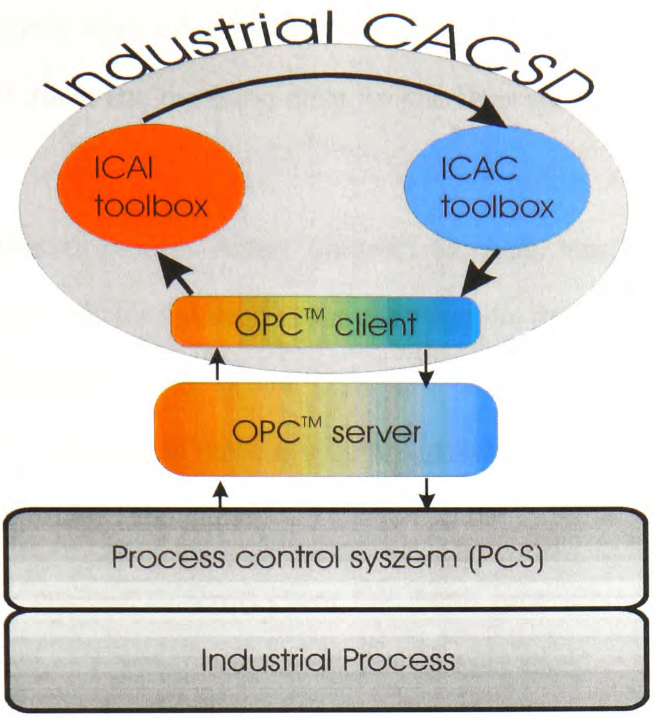


Figure 7-6 Current state of the Industrial CACSD scheme

The validation of the OPC™ client is described in Chapter 8.4 with detailed information about the process control system.

7.4 Prototype Limitations

The *ICAC* prototype realisation is restricted to the MATLAB™/SIMULINK environment and can only handle process models generated from the identification toolbox *ICAI*. The nonlinear MIMO CD block is still under development. The project settings are not hereditary at the moment. Because of this only the process personnel level has been completely realised for the realised *ICAC* CD blocks.

7.5 Conclusion of this Chapter

The procedures and methods proposed in the preceding chapters have been realised by the presented prototype software. The general considerations for the prototype development were inherited from the *ICAI* toolbox to define a clear frame for this project and make the handling easy for the user within the *Industrial CACSD* framework.

The *ICAC* (Industrial Computer Aided Control) blockset has been described and offers a new framework for control system design in the process industry. It is integrated into SIMULINK and offers standard design paths to the design of singlevariable and multivariable control systems utilising GUIs which are adapted to the user's knowledge and capabilities.

An OPC™ (OLE for Process Control) client has been programmed as a gateway to process control systems to allow easy access to process data.

8 Prototype Validation

This chapter gives a brief summary of the tests carried out in order to verify and validate the proposed novel approach. The novel approach, in this context, refers to the procedures and methods that have been implemented into the *ICAC* prototype as outlined in Chapter 7

8.1 Description of the Test Procedure

Various tests have been carried out in the control laboratory at the Fachhochschule Hannover with the laboratory staff and students. The properties of the laboratory plant involvement of investigation is summarised in Table 8-1. The laboratory air-conditioning plant (LKR) will be described below.

Table 8-1 Application example

Process	LKR (laboratory air-conditioning plant)
Output (Control) Variables	o temperature o humidity
Input Variables	o heating voltage o humidity voltage o fan voltage (air flow)

8.1.1 Laboratory air-Conditioning Plant (LKR – Labor Klima Regelung)

8.1.1.1 General Information

The LKR is a climatic chamber control process implemented at laboratory scale and intended for educational purposes in the area of automatic control and general engineering studies associated with climatic thermodynamics. The LKR was designed and built as part of a European TEMPUS - Projects JEP 4298/4299 in a collaboration between Vytautas Kaminskas University, Lithuania, University of Ljubljana, Slovenia, University of Glamorgan, United Kingdom, HTWKL Leipzig, Germany and Fachhochschule Hannover, Germany during the years 1992-94.

The plant has two controlled variables: air temperature and relative humidity. The corresponding actuators are a heater and a nebulizer. Both variables are intrinsically cross coupled and an increase of temperature decreases the relative humidity and conversely humidifying decreases the temperature of the air. The plant therefore represents a true multivariable system with two inputs and two outputs. Nevertheless, it can also be used as a single variable system if only one of the two control paths are taken into account.

According to educational objectives, the basic hardware configuration shown in Figure 8-1 was developed: a fan was used for conveying the air to be heated and humidified, to the chamber. A heating coil acted as the actuator for affecting the temperature control and a nebulizer was used as the process inputs.

The heating coil was situated in a glass tube for which a maximum heating power of approximately about 25W was chosen and was custom made by the Institute Jozef Stefan in the Faculty of Electrical Engineering, Ljubljana, Slovenia. The nebulized water, introduced downstream, was delivered by a nebuliser operated by compressed air which was supplied by a small compressor (20W) and was manufactured by the Institute of Electronics and Vacuum Technology, Ljubljana, Slovenia (It was taken from a medical system meant for children and people with breathing problems that the Institute produces for inhalation). A mixture of the heated and humidified air entered the mixing chamber, in which the final temperature and humidity were measured and which represented the controlled variables and process outputs (ϑ_3, φ_3) respectively. In addition, intermediate measurements of temperature and humidity were taken of the air inlet after the ventilator (ϑ_1, φ_1) and after the heating coil (ϑ_2, φ_2). The three humidity and temperature sensor blocks were manufactured by the Institute of Electronics and Vacuum Technology, Ljubljana, Slovenia, type: HTT2. This enabled more insight into the process and the implementation of cascade and feedforward control.

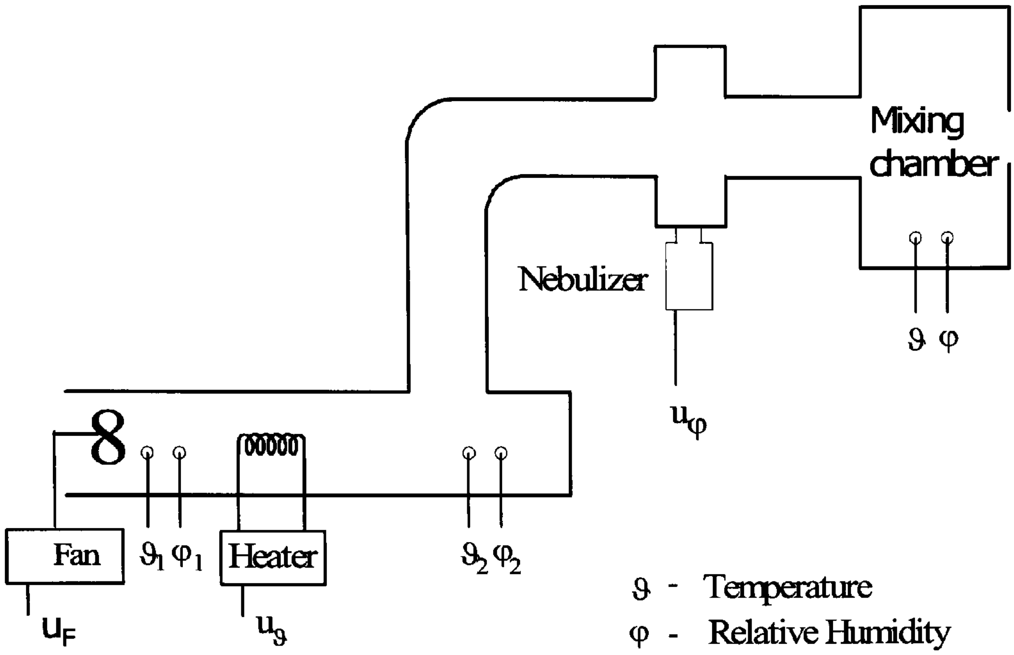


Figure 8-1 Process scheme of the laboratory air-conditioning plant (LKR)

The block diagram of the climate process is shown in Figure 8-1. Time constants of the actuators were found to be less than 1 s and were therefore considered to be negligible and as such are not shown in the block scheme. The process is highly nonlinear (in the static sense) firstly because of the quadratic relationship between heating power and the input voltage, secondly because of the nonlinear characteristic of the nebulizer and thirdly because of the complicated relationship between temperature and humidity in accordance with the Mollier diagram.

8.1.1.2 Technical Specifications of the LKR

The LKR, as shown in Figure 8-2, is built into a 19" rack compact casing which includes the process, all sensors and transducers, actuators and a power supply. All I/O signals are brought to the front panel and corresponding values are shown on LCD displays. External connections include 220V power socket at the rear. Two input sockets for the actuators and six output sockets (two per measuring point – temperature and relative humidity) are included.

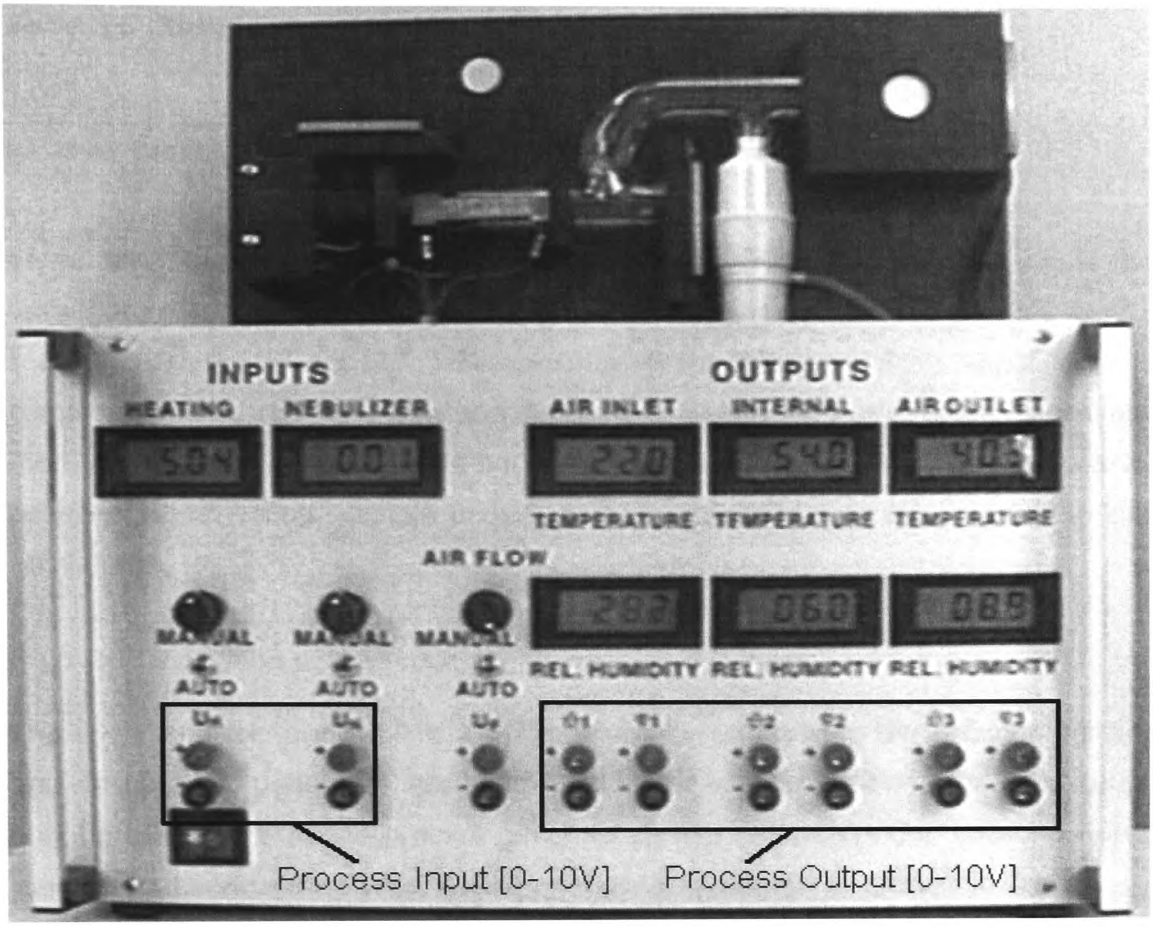


Figure 8-2 Front panel of the laboratory air-conditioning plant (LKR)

Table 8-2 Input Signals

Heating	u_ϑ	0 V – heater off	10 V – full heating power (approx. 25W)
Humidifying	u_φ	0 V – compressor off	10 V – compressor operates at max. pressure
Airflow	u_F	0 V – minimum speed	10 V – full speed

There are three input signals but only two of them are used as process inputs: heating voltage u_ϑ and voltage on the nebulizer u_φ, which determines the intensity of humidification. The third input signal u_F determines the airflow and should be fixed during the experiment unless it is used as a disturbance to the process.

Table 8-3 Output Signals

Temperature	0 V – 0 °C	10 V – 100 °C
Relative Humidity	0 V – 0 %RH	10 V – 100 %RH

There are six output signals corresponding to three pares of measuring points for temperature and relative humidity. Each output signal is accessible by connection to sockets marked "+" and "-" . Temperature and relative humidity signal have a common ground, meaning that sockets for the temperature and relative humidity marked with "-" are internally connected at each measuring point. The relationship between signal voltage (single ended) and corresponding quantity is listed in the above table.

8.1.1.2.1 Operation of the Process

The cover and the upper plate were removed by unscrewing the securing screws and the process panel was lifted and fixed in the vertical position. The plant could only be operated with the process panel set to the vertical position because water leakage could occur otherwise, causing danger and damage to electronic circuits.

8.1.2 Experimental Set-up

The experiments were carried out utilising a separate tool ('Dora für Windows') for test signals and data acquisition. (In the final version of *Industrial CACSD* data acquisition will be accomplished by means of an interconnected process control system with e.g. OPC™ server/client function or in the case of stand alone applications by making direct use of AD/DA process interfaces from within SIMULINK.) The required process models were identified with the MATLAB™ toolbox *ICAI*.

The application of *ICAC* for the control system design are shown for a nonlinear multivariable process (LKR) a linear and nonlinear single variable control system and a linear and nonlinear multivariable control system. The results of the control system design are presented in the form of SIMULINK blocks within *ICAC* windows and the results of the implemented control systems are shown in signal diagrams (Figure 8-7 – 8-15).

8.1.3 Identification of the LKR Process Model using the *ICAI* toolbox

The strong couplings between temperature and humidity make the MIMO modelling necessary. The experiments on the process were performed within the operating range of the process. The air flow was fixed at 10V such that the two inputs, heater voltage and nebulizer voltage, influenced temperature and humidity respectively.

Figure 8-3 and Figure 8-4 show the static characteristics of the laboratory air-conditioning plant within the operating range. In both cases one of the input variables (heater voltage or nebulizer voltage) was fixed at 5 Volt and the other input variable was changed from 0-10 Volt after reaching the settling point.

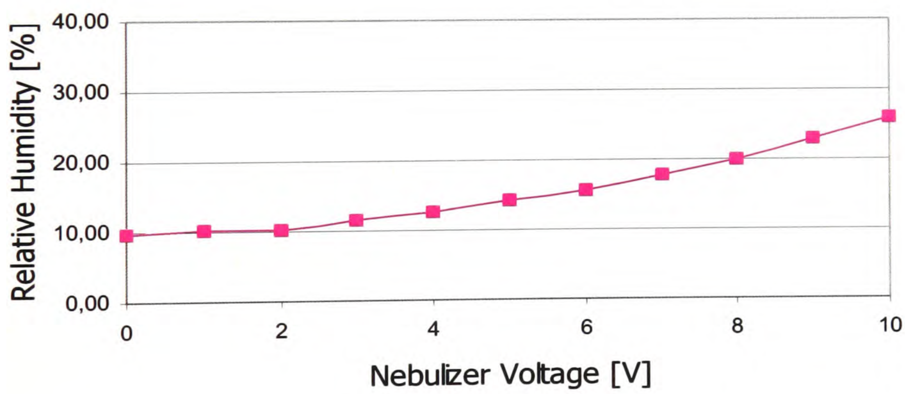


Figure 8-3 Static Characteristic of the LKR (ICAI hum static)
Heater voltage: 5 Volt, Nebulizer voltage: 0-10 Volt

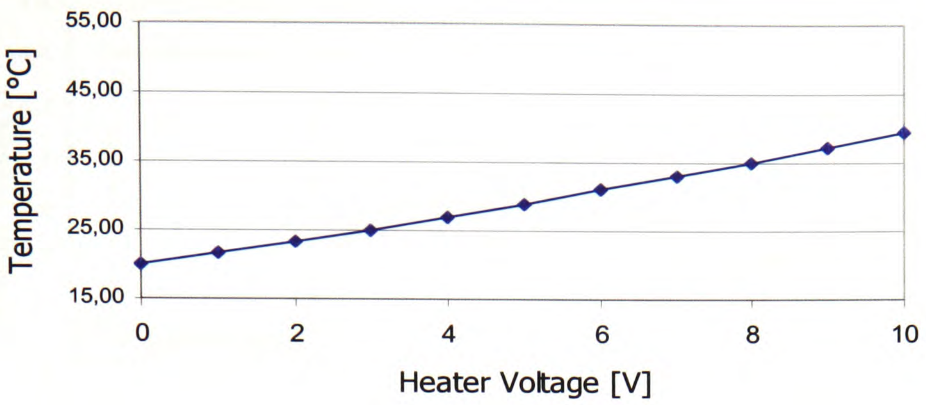


Figure 8-4 Static Characteristic of the LKR (ICAI temp static)
Nebulizer voltage: 5 Volt, Heater voltage: 0-10 Volt

The diagrams in the above figures depict the relatively small operating range of the air-conditioning plant, Table 8-4 summarises the operating range of the heater voltage and nebulizer voltage numerically.

Table 8-4 Operating range the LKR

	Operating Range
Temperature	20 – 30 °C
Humidity	10 – 25 %

The identification experiment was again carried out within the operating range. Figure 8-5 shows the changing temperature in the mixing chamber with respect to the changes in the set points of the heating voltage and nebulizer voltage and Figure 8-6 shows the changes of the humidity in the mixing chamber with respect to the changes in the set points of the heating voltage and nebulizer voltage.

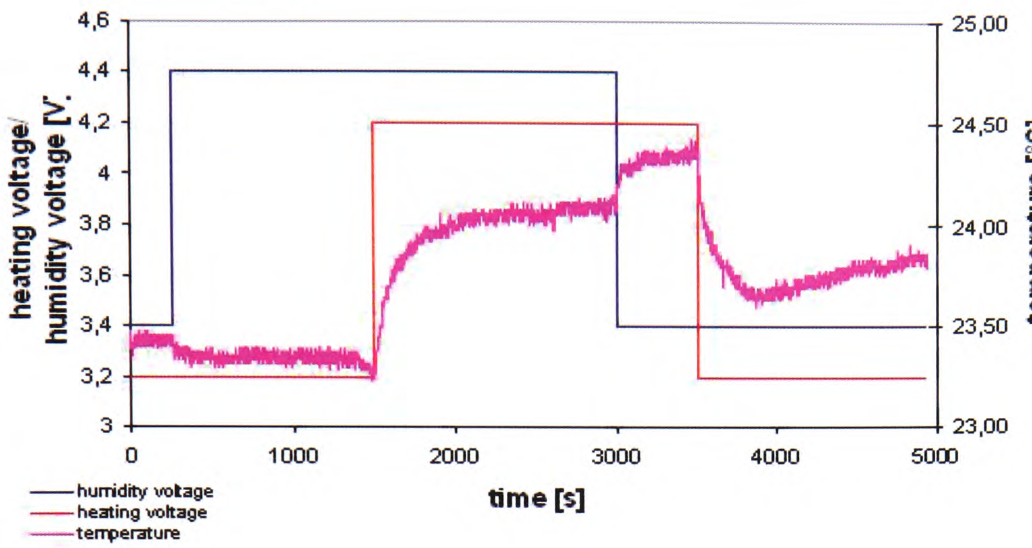


Figure 8-5 Identification experiment of the LKR, step response of temperature

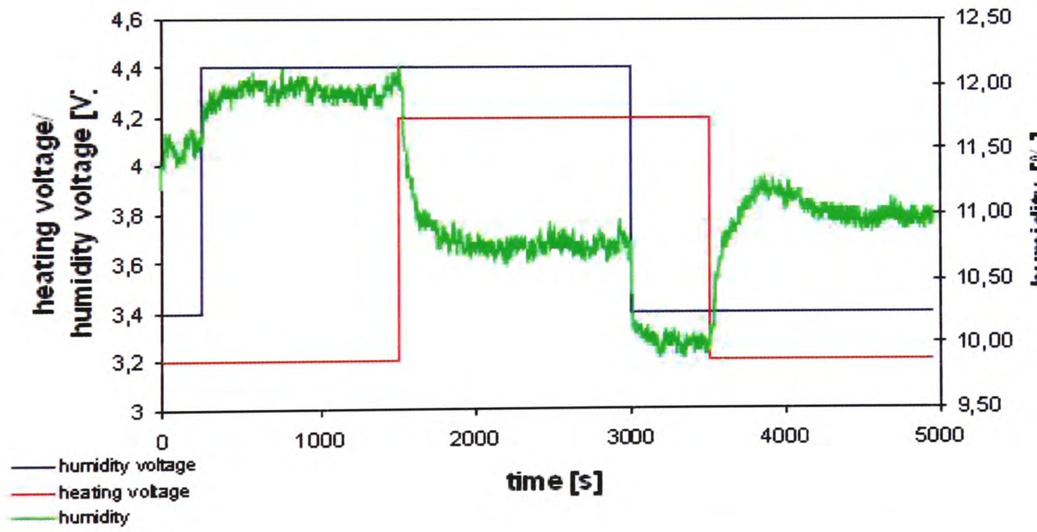


Figure 8-6 Identification experiment of the LKR, step response of humidity

The linear dynamic part has been identified using the linear dynamic MIMO ID block from the *ICAI* toolbox and the static characteristics have been generated using the static *ICAI* SISO ID blocks. Figure 8-7 presents the identified nonlinear multivariable plant model (as SIMULINK blocks) which was used for all simulations in the following sections.

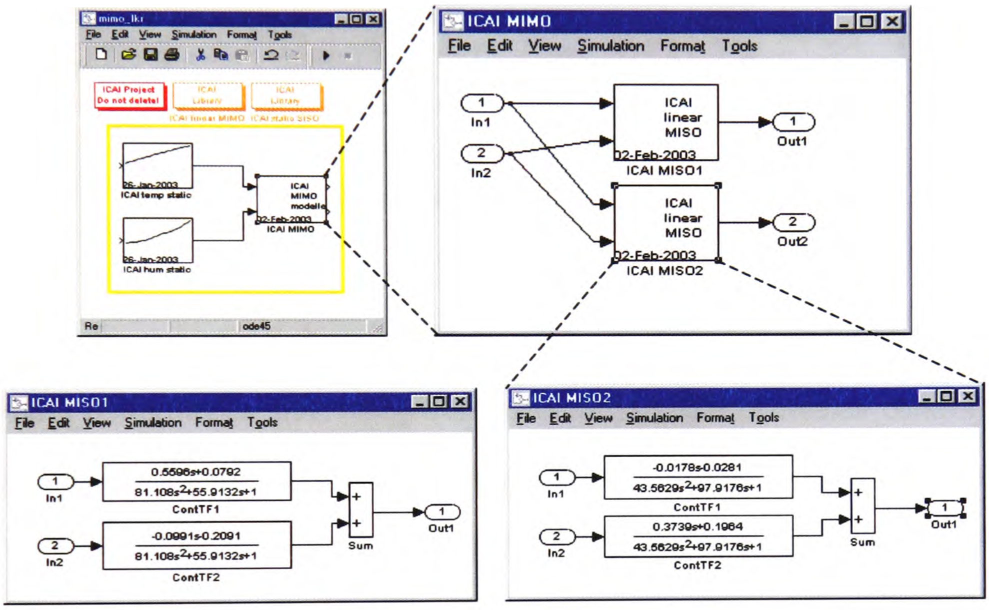


Figure 8-7 Nonlinear MIMO process model of LKR identified with ICAI

8.2 SISO Control System Design for LKR Plant (SISO Control)

The nonlinear MIMO *ICAI* process model (Figure 8-7) identified above, has been used for the design of linear and nonlinear SISO controllers. The main process model parts (G_{11} and G_{22} of Figure 8-8) have been selected for the design of an independent SISO PID control system (PID_{11} and PID_{22} of Figure 8-8). The set point variables (temperature and humidity) used for the identification were modified by reducing the offset. These modified set point variables were used for the control system design procedure.

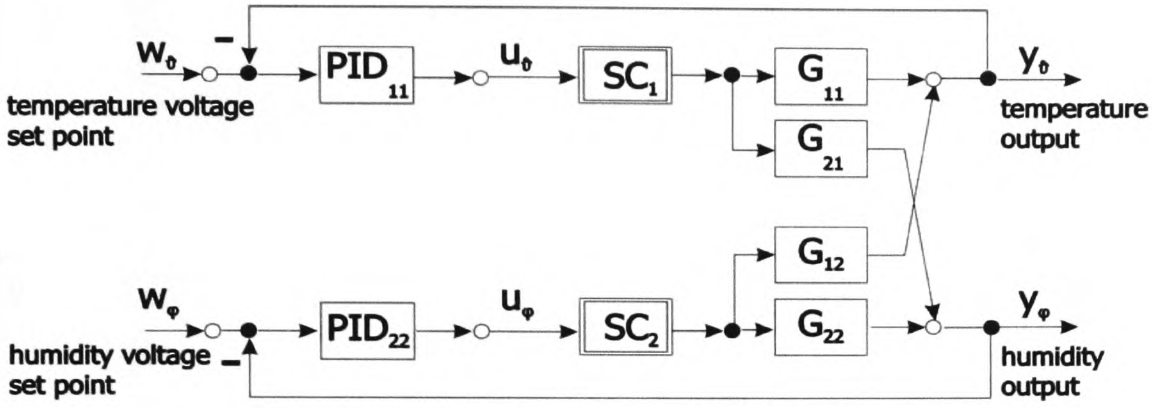


Figure 8-8 Nonlinear MIMO process model with main linear SISO PID controller

The *ICAC* default settings of the linear SISO CD block offer the user modification of the control performance only, using the sliders for stronger or weaker control action in the simulated closed-loop control system. In this way satisfactory results were obtained easily. The resulting vector of the state feedback gains, K with r=10 came out to be K=[0,31 11,51 16,46] and the approximation of the PID controller obtained by numerically minimising the residuals between the optimal state variable feedback controller and the corresponding PID controller resulted in the parameters K_p=2,64, K_i= 0,19, K_d= 0.

Figure 8-9 shows the simulated step responses in the *ICAC* design window of the laboratory air-conditioning process, composed of *ICAC* SISO CD blocks.

The simulation shown in Figure 8-9 a) illustrates the step response of the optimal control system. Figure 8-9 b) shows the comparison between the process performance with the optimal control system and the corresponding approximated PID controller. Figure 8-9 c) shows the resulting singlevariable control systems.

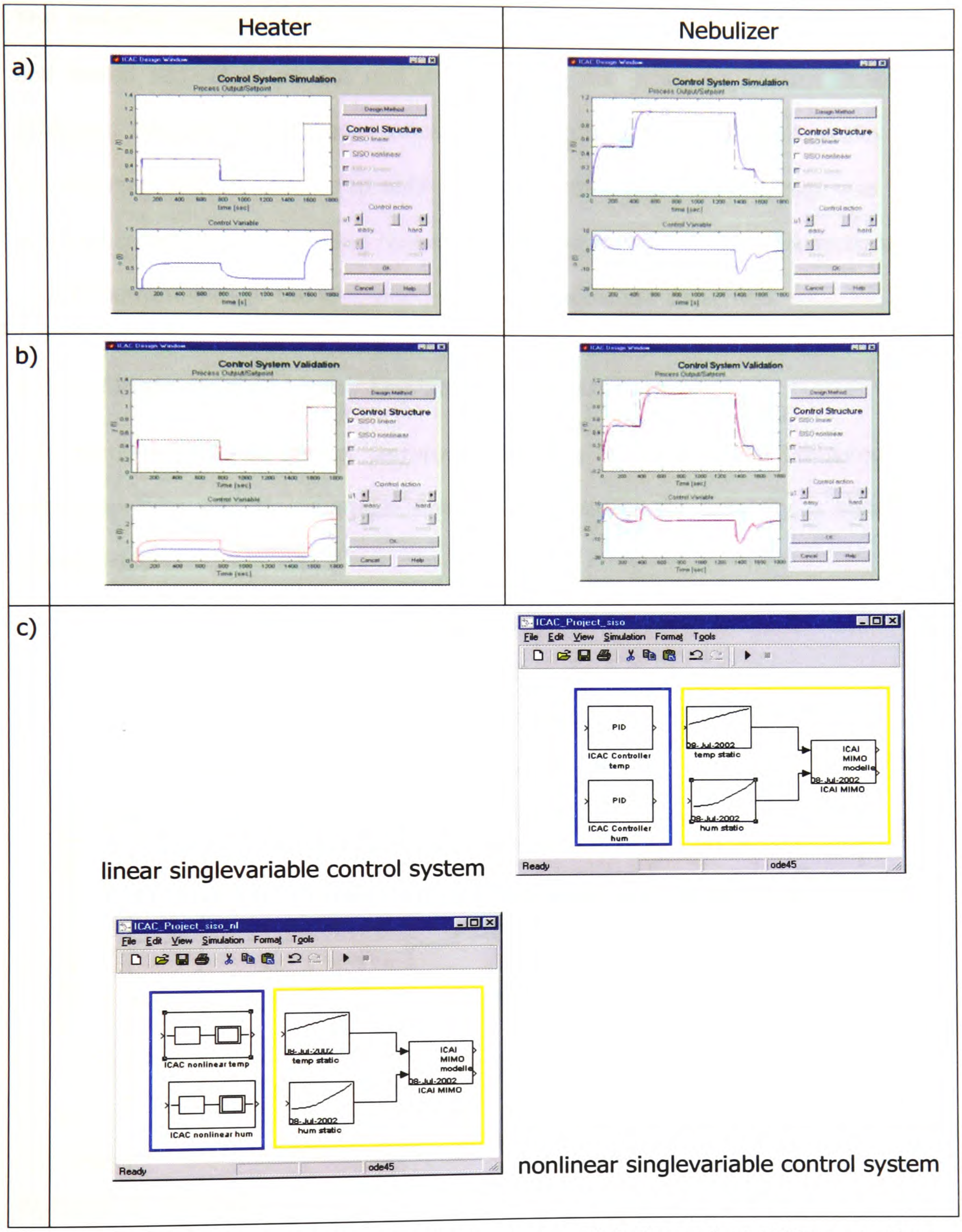


Figure 8-9 Laboratory climate control system with independent SISO controllers

The designed singlevariable linear and nonlinear SISO controllers were simulated and tested separately on the real process using for prototype control *DORA for Windows*. Because of the bigger influence and the nonlinearity of the humidity only the nebulizer with its associated data was considered. The heater voltage was fixed at 3 Volt by which the operating range has been postponed. In the next sections the control performance of the simulation and real process are presented.

8.2.1 Singlevariable Linear Control System (Linear SISO)

The simulation of the control performance of the humidity in the simulated LKR controlled with one singlevariable linear PID controller for humidity is shown in Figure 8-10.

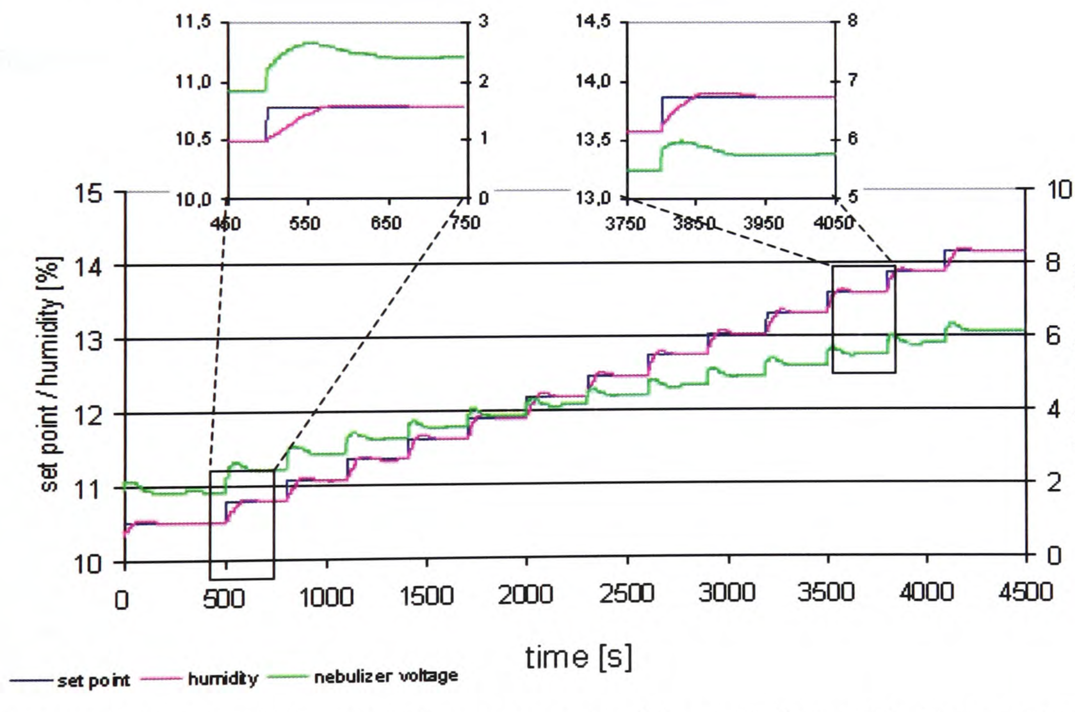


Figure 8-10 Humidity of LKR process model controlled with linear SISO PID controller

The control performance of the humidity of the real LKR plant controlled with one singlevariable PID controller is depicted in Figure 8-11.

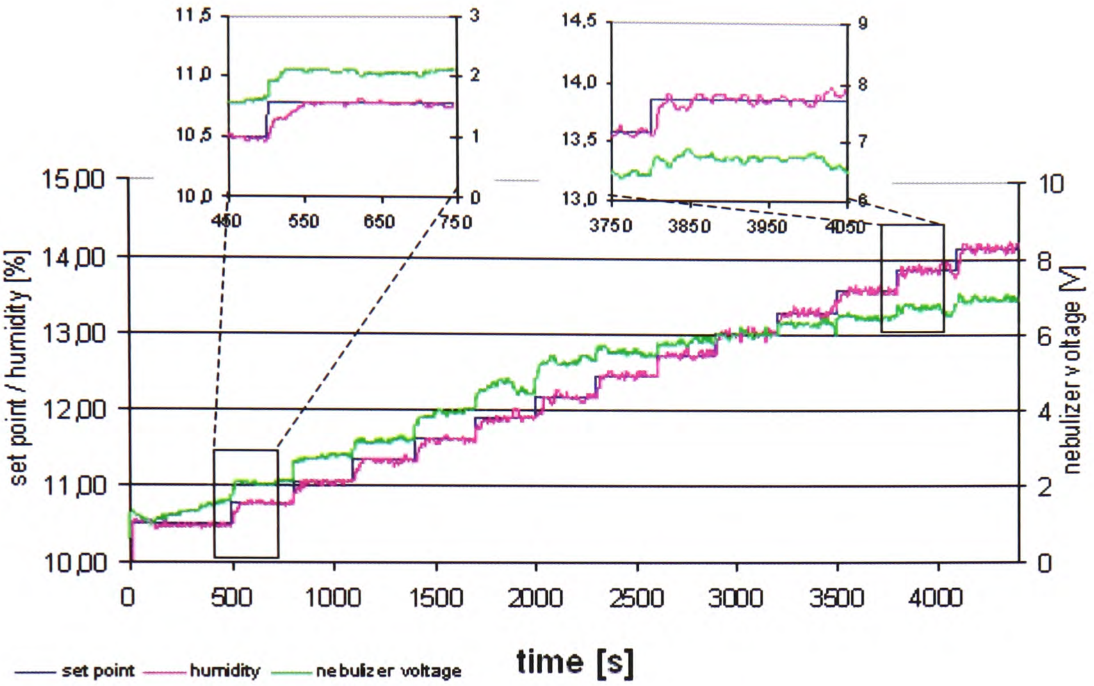


Figure 8-11 Humidity of real LKR plant controlled with linear SISO PID controller

8.2.2 Singlevariable Nonlinear Control System (Nonlinear SISO)

The control performance using nonlinear SISO control for humidity is shown in Figure 8-12.

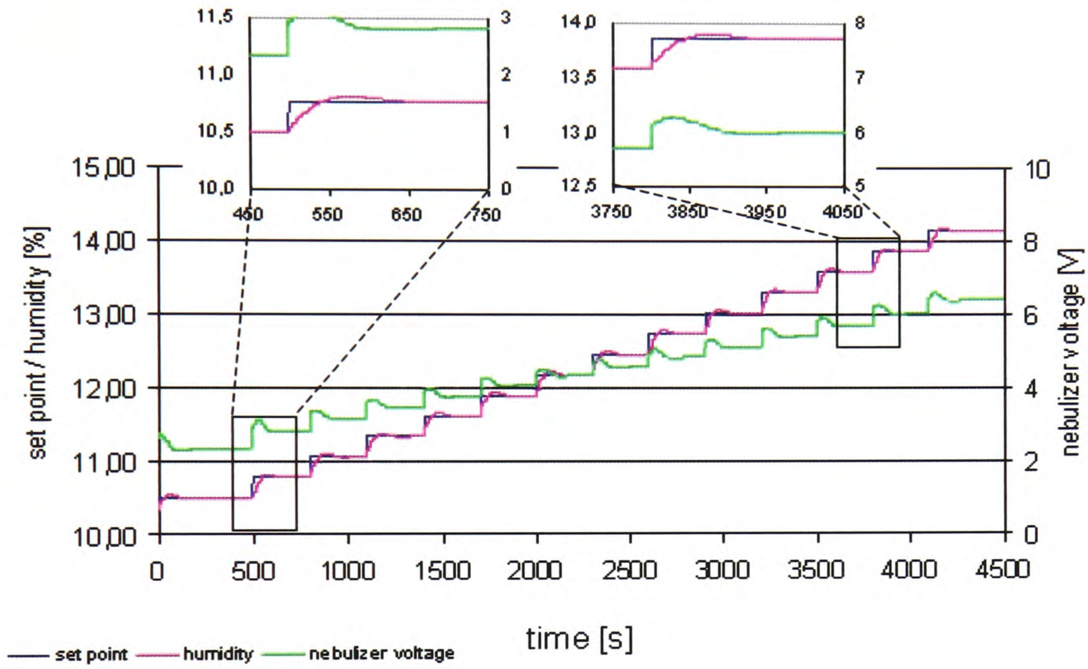


Figure 8-12 Humidity of LKR process model controlled with nonlinear SISO PID controller

Figure 8-13 depicts the control performance for humidity at the LKR plant controlled by the nonlinear SISO controller.

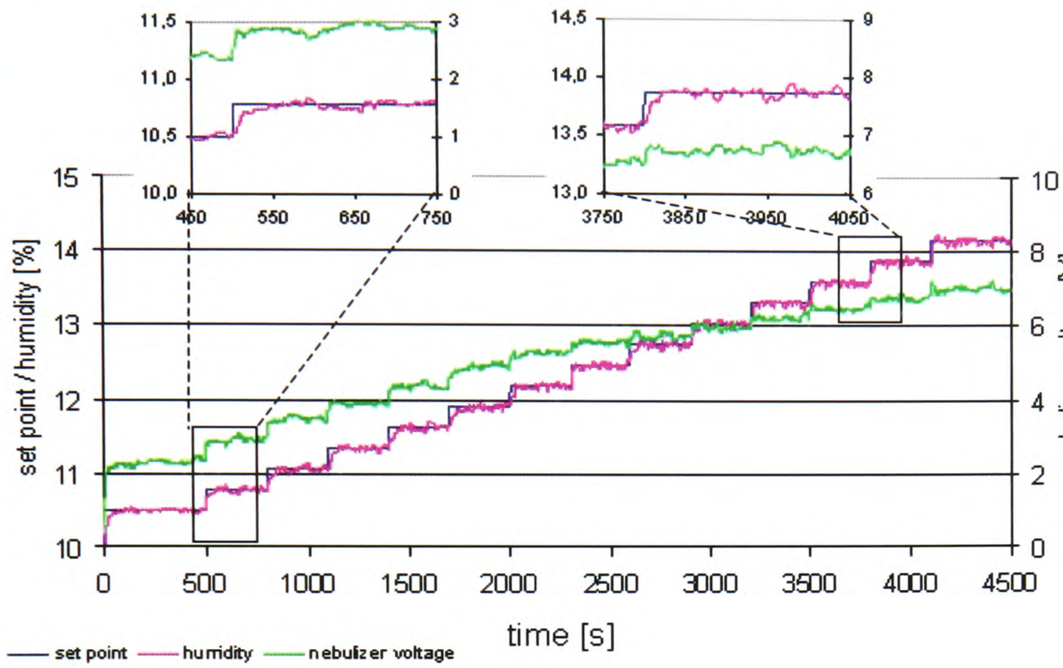


Figure 8-13 Humidity of real LKR plant controlled with nonlinear PID SISO controller

8.2.3 Linear versus Nonlinear Control and Simulation versus Real Process Experiments

The resulting singlevariable control systems for the humidity were considered by conducting simulations and then tested on the real (physical) process for comparison. The linear SISO control shows different control performances at for example 10% and 14% relative humidity. The nonlinear control on the other hand shows similar control performance at 10% and 14% relative humidity. The effects of the nonlinearity on the process are compensated by the inverted static characteristic of the nonlinear controller, within the linear control these effects are discernable. The nonlinear controller controls the change in the set point quicker than the linear controller.

The simulation and the real plant have similar control responses. The real plant control shows more oscillations due to small disturbances (input air) and detuned control. It has to be stressed that the fact that control performance of the simulation and the real process are similar shows how the simulation results will predict the control performance of the real plant.

8.3 MIMO Control System Design for the LKR Plant (MIMO Control)

The nonlinear MIMO *ICAI* model identified (Figure 8-7) has also been used for the design of linear and nonlinear multivariable control systems.

The *ICAC* default settings for the linear MIMO CD block support the user considerably, providing default control parameters. In this way satisfactory results have been gained easily. The resulting vector of the state feedback gain K with $\underline{R} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$ came out to $\underline{K} = \begin{bmatrix} 0,16 & 4,54 & 14,34 & 0 & 0 & 18,97 \\ -0.07 & -7,33 & -5.17 & 0,51 & 11,97 & -1,42 \end{bmatrix}$

and the PID control system resulted in the parameters

K_{p11}= 4,53	K_{p11}= 0	K_{p11}=-7,33	K_{p11}= 11.97
K_{i11}= 0,16	K_{i11}= 0	K_{i11}= -0,07	K_{i11}= 0,51
K_{d11}= 14.34	K_{d11}= 18.97	K_{d11}= -5,17	K_{d11}= -1,42

Figure 8-14 a) shows the simulated step responses in the *ICAC* design window of the laboratory air-conditioning process controlled with a linear *ICAC* MIMO CD block. The accepted control performance of the simulated closed loop controlled with the optimal control system was automatically approximated by a PID control system and transferred into the linear MIMO control block, as shown in Figure 8-14 b).

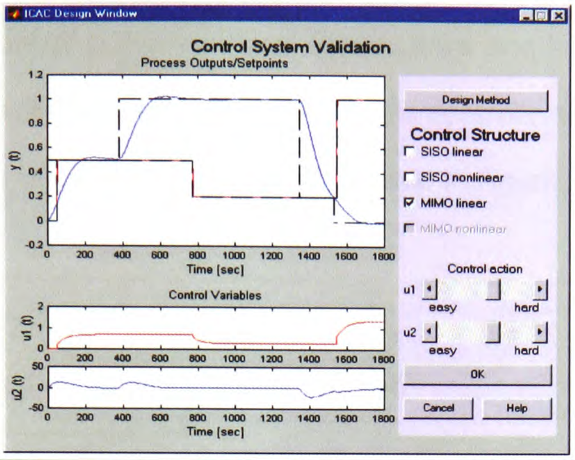
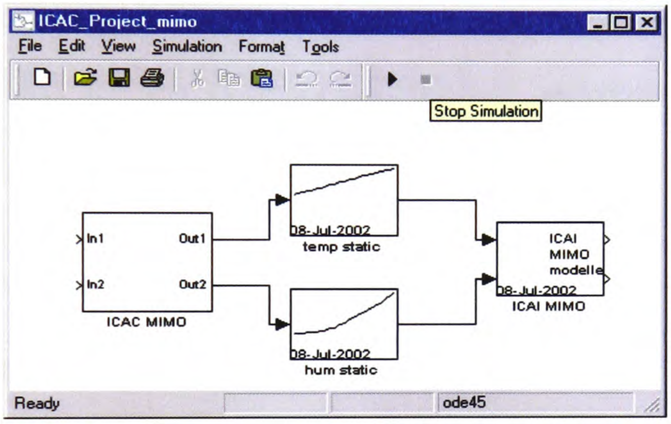
	Heater and Nebulizer
a)	
b)	
	linear multivariable PID based control system

Figure 8-14 LKR model composed from MIMO controllers

The control system that has been designed to Figure 8-14, was simulated and tested on the real process. In order to show the superiority of the multivariable control system the control performance of the multivariable linear PID control system with decoupling is presented and compared to the control performance of two independent linear SISO PID controller without decoupling, as shown in Figure 8-8.

8.3.1 Two Independent Linear SISO Controller

The simulation of the control performance of temperature and humidity of the LKR model controlled by two independent linear SISO PID controllers, which have been designed and are shown in Figure 8-8 is then presented in Figure 8-15.

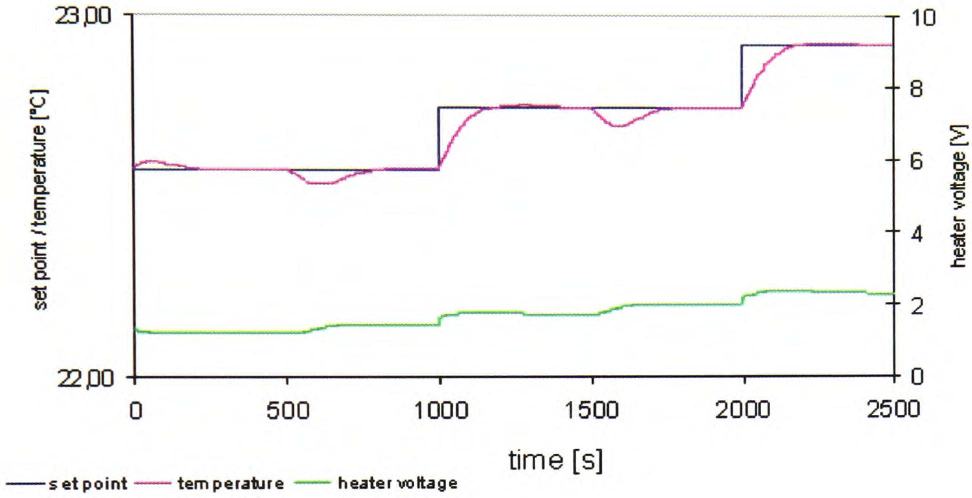


Figure 8-15 a) Temperature control

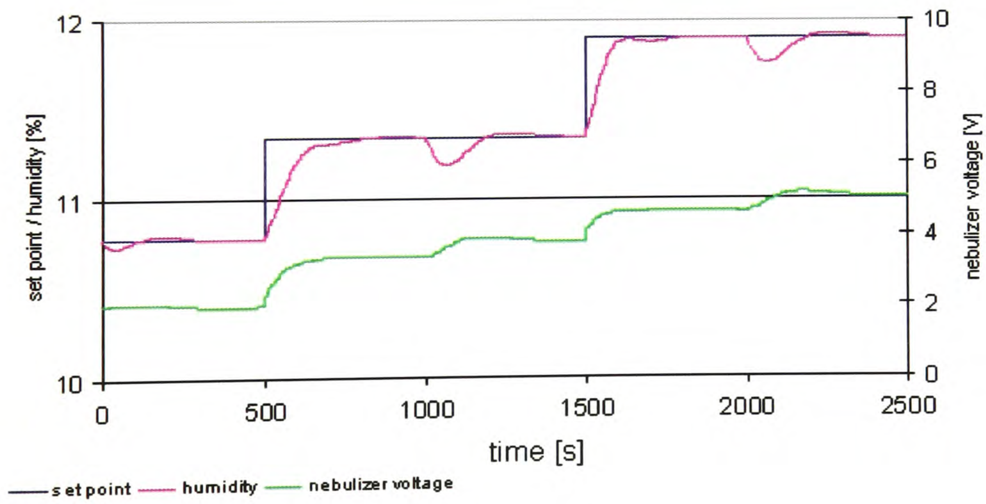


Figure 8-15 b) Humidity control

Figure 8-15 Control performance of the simulated LKR process model with two independent linear SISO controllers

The control performance of temperature and humidity of the real LKR plant controlled by two independent linear SISO PID controllers is presented in Figure 8-16.

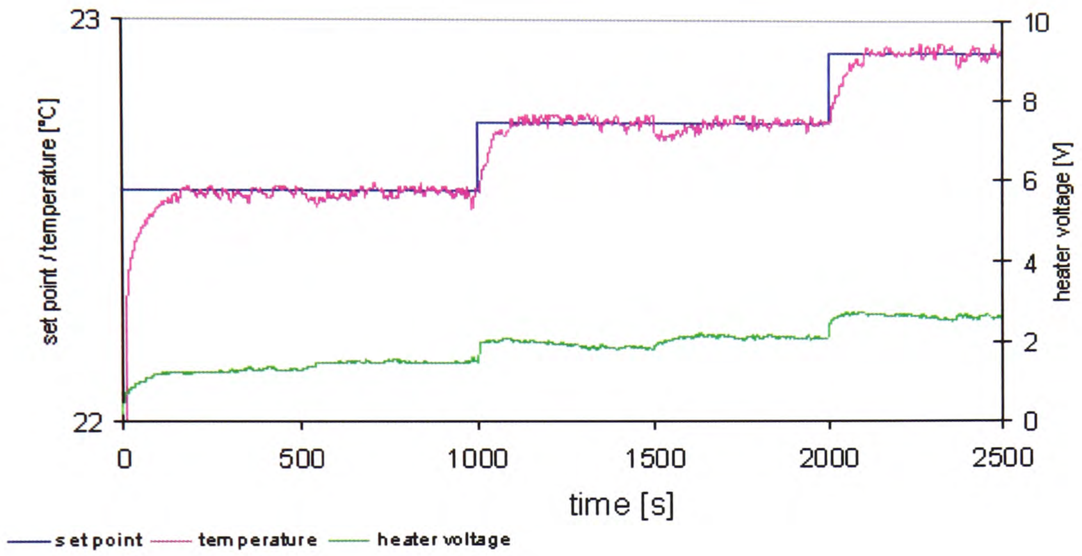


Figure 8-16 a) *Temperature control*

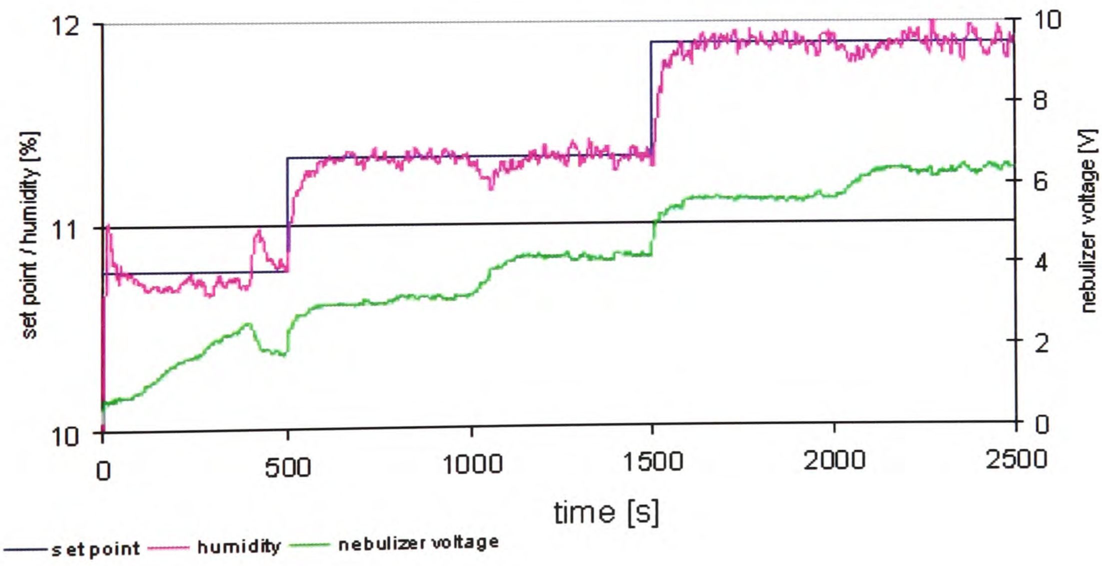


Figure 8-16b) *Humidity control*

Figure 8-16 *Control performance at real LKR plant with two independent linear SISO controllers*

8.3.2 Linear MIMO Control

The simulation of the control performance of temperature and humidity of the LKR model controlled with the linear MIMO control system is presented in Figure 8-17 which has been designed and is depicted in Figure 8-14.

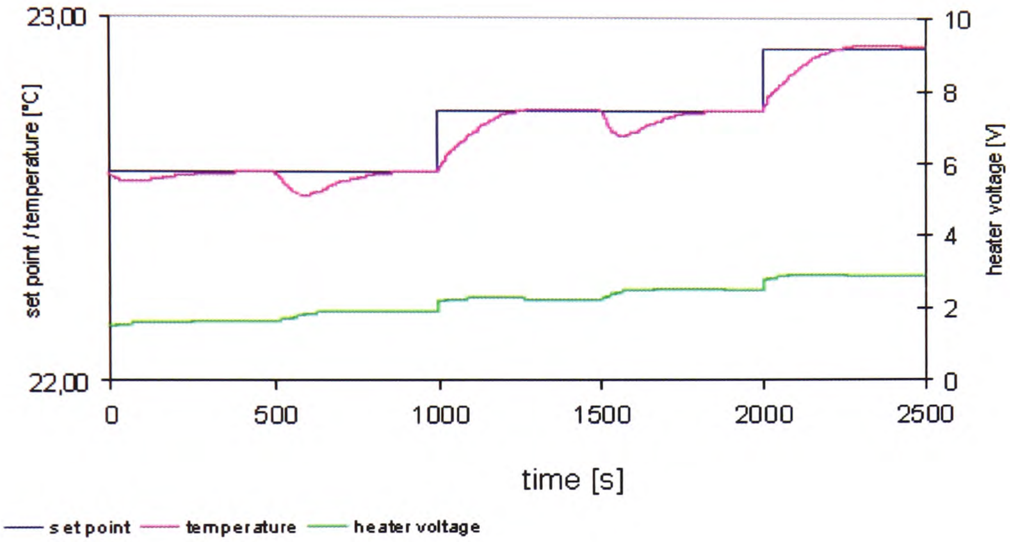


Figure 8-17 a) Temperature control

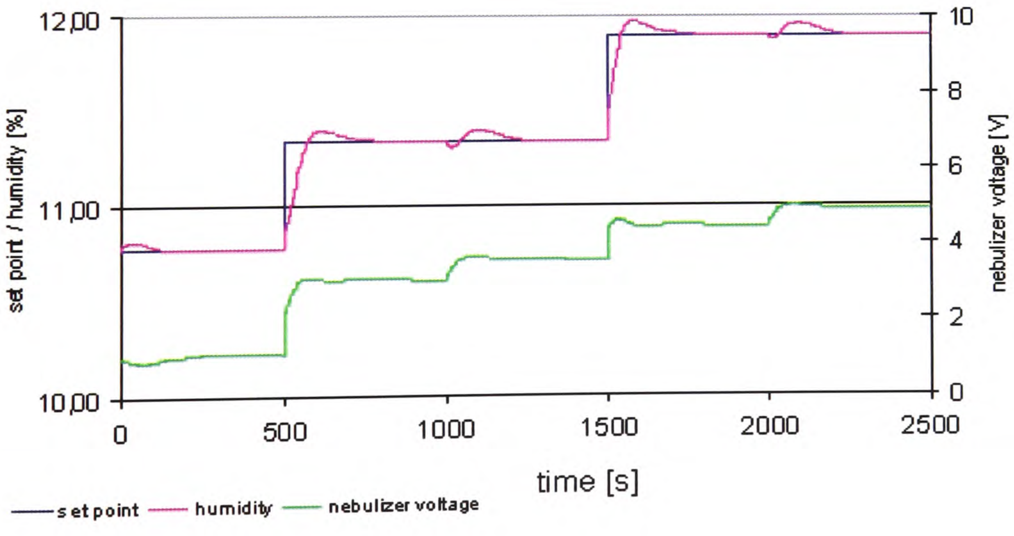


Figure 8-17 b) Humidity control

Figure 8-17 Control performance at simulated LKR process model with decoupling linear MIMO control

The control performance of temperature and humidity of the real LKR plant controlled with the linear MIMO control system is presented in Figure 8-18.

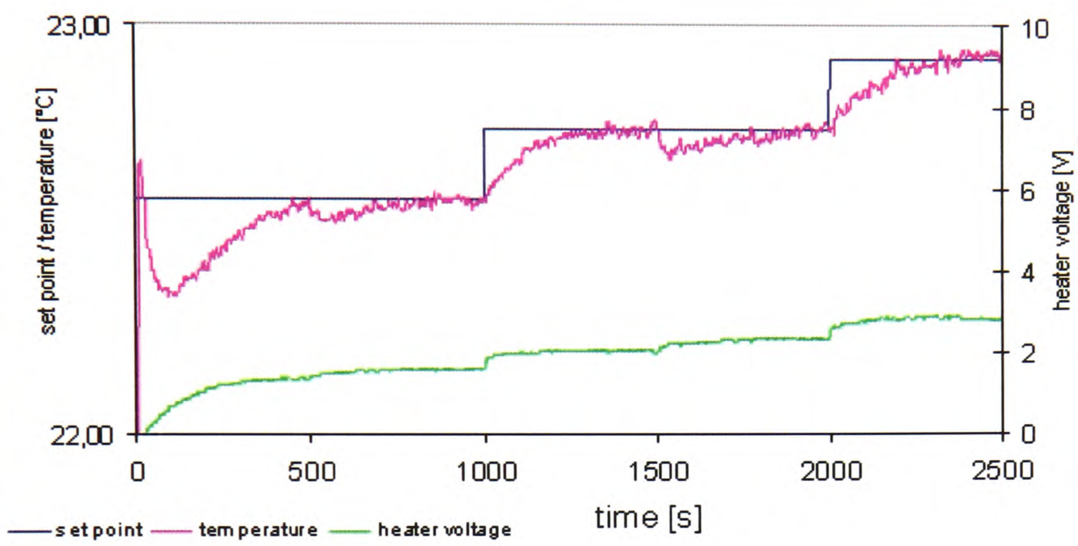


Figure 8-18 a) Temperature control

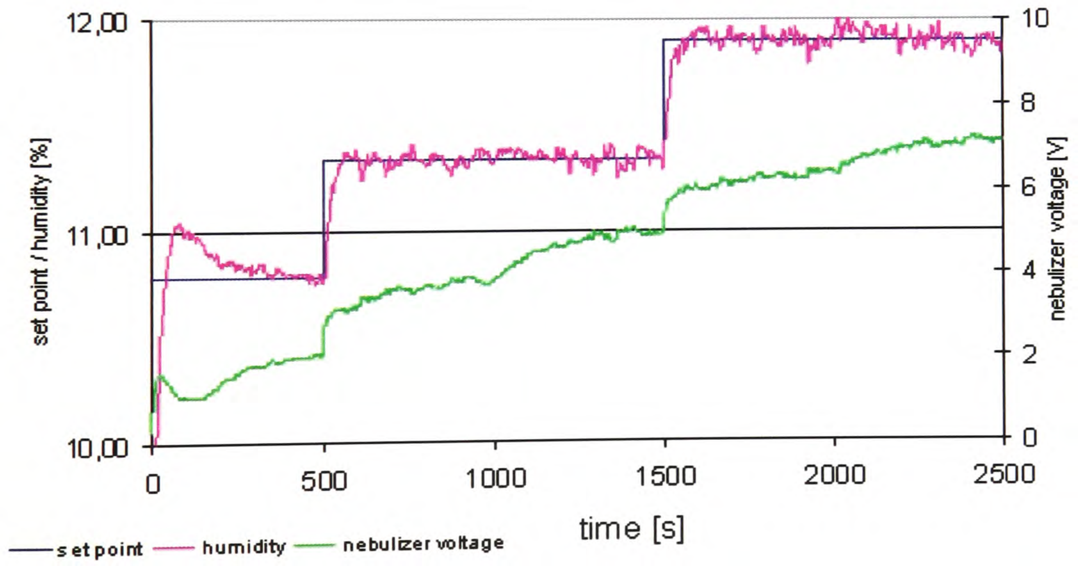


Figure 8-18 b) Humidity control

Figure 8-18 Control performance at real LKR plant with decoupling linear MIMO control

8.3.3 Independent Linear SISO Controllers versus Decoupling Linear MIMO Control

In order to show the superiority of the multivariable control system the control performance of the multivariable linear PID control system with decoupling is compared to the control performance of two independent linear SISO PID controllers without decoupling. The coupling effects of the laboratory air-conditioning process are clearly recognisable in the control performance with two independent linear SISO controllers at all step changes of the set points variables (time 500, 1000, 1500, 2000). With decoupling linear MIMO control these effects are considerably reduced especially for the humidity performance.

These improvements are clearly risible in the simulation results and also recognisable in the experiments at the real LKR plant.

8.4 Tests in Industrial Environment

It is intended to test in conclusion the *ICAC* prototype in an industrial environment. To realise an industrial experimental set-up in the laboratory for this project, the industry partner ABB Automation supported the installation of a process control system (Freelance 2000®) at the Fachhochschule Hannover. Freelance 2000® is a scalable process control system which is divided into an operator level and a process level. The operator level includes traditional process control system functions such as operation and observation, archiving, and logs, as well as trends and alarms. Loop and logic control functions are processed, usually in proprietary process stations.

The operator stations use standard or industrial PC hardware running the Windows NT® operating system. One engineering station and several operator stations can be installed at the operator level. The engineering station is used to configure and commission the system. At the process level, a Freelance 2000® system can consist of several process stations or FieldControllers that can be extended with I/O units. The industrial system architecture is shown in Figure 8-19.

In the laboratory the Freelance 2000® operator station software DigiVis and the engineering station software DigiTool were installed on one workstation. Furthermore a process station was connected to the laboratory air-conditioning plant (LKR), described in Chapter 8.1. A Freelance project was configured using the engineering station to run and operate the air-conditioning plant.

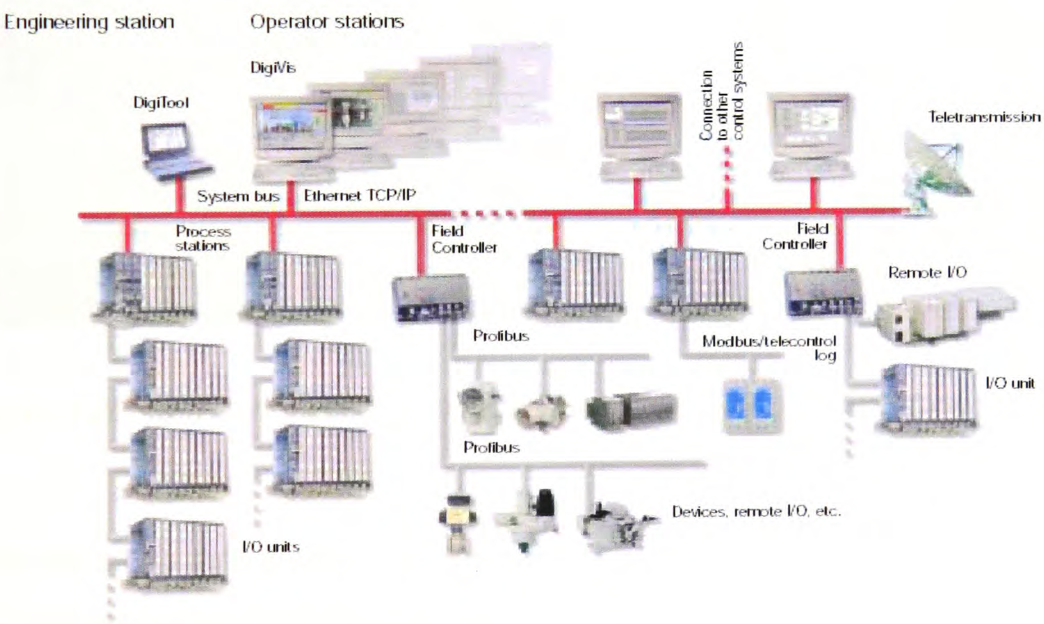


Figure 8-19 Freelance 2000® system architecture

To allow easy access to process data in the process station Freelance 2000® provides an OPC™ (OLE for Process Control) server. The programmed OPC client provides services for the efficient reading and writing of data between the client application, here the *ICACSD* PC with *ICAI* and *ICAC*, and the process control device, here Freelance 2000® .

The configuration of the process control system Freelance 2000® connected to the laboratory air-conditioning process is shown in Figure 8-20.

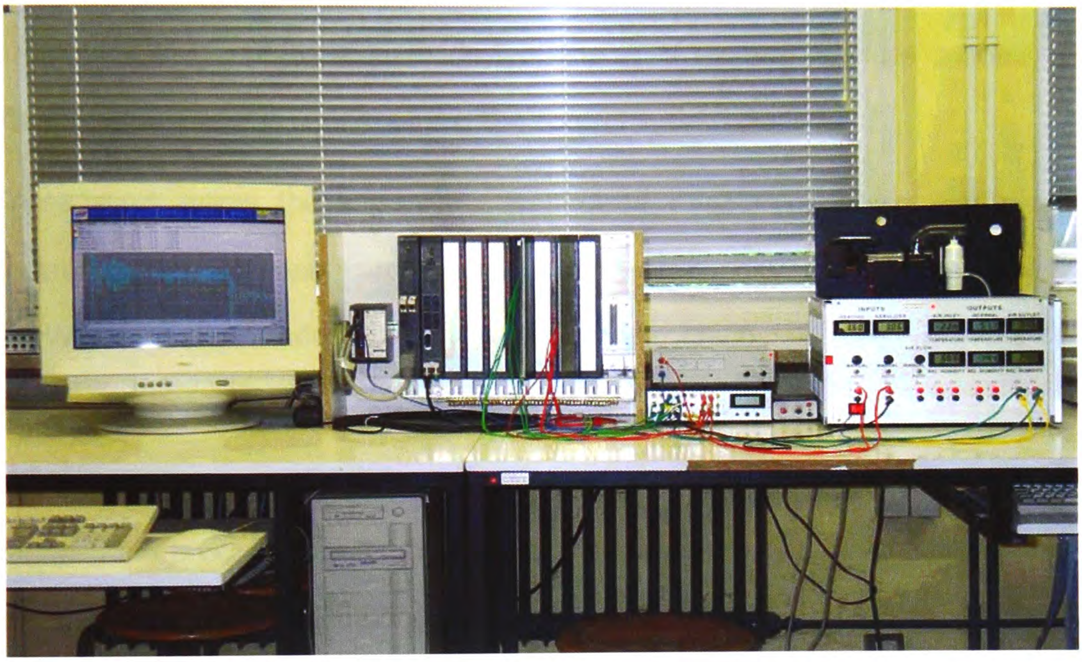


Figure 8-20 Freelance 2000® connected to the laboratory air-conditioning process

With the online data access using the OPC client, test signals can be written to the process and process data can be read out from the plant floor. In the current state the direct implementation of the designed control system into the Freelance 2000® process station is not allowed due to safety considerations.

8.5 Conclusion of this Chapter

In the context of the collaborative research project, this work provides an *Industrial CACSD* scheme (Chapter 4) and an appropriate approach to control system design that can even be applied by inexperienced users. These developments extend the identification module *ICAI* (Industrial Computer Aided Identification) (Körner, 1999). In this context the *Industrial CACSD* scheme defines the overall strategy for the computer based control system design, the *ICAI* toolbox provides process model identification and the *ICAC* toolbox, the control system design.

These steps follow the improved standardised *CACSD* procedure. The prototype control system has been tested on a laboratory air-conditioning process using DORA for Windows or as industrial control system the Freelance 2000® process control system

The application of the *ICACSD* system with *ICAI* and *ICAC* to the laboratory air-conditioning process has demonstrated the applicability of the prototype software and the incorporated methods. Some prototype validation tests were conducted on a laboratory scale air-conditioning process which was described in section 8.1.1. The identification of the process model was described in section 8.1.2. The control design procedure for linear and nonlinear SISO control systems and their validation test results were outlined and a comparison was made in sections 8.2. The control design procedure for linear MIMO control systems, the results of the validation test and the comparison of independent linear SISO controllers versus decoupling linear MIMO control were outlined and discussed in section 8.3.

As preparation for the transfer of the strategy presented in this thesis to an industrial environment an OPC™ interface was programmed for the process control system Freelance 2000® from ABB Automation. At the present state this allows data acquisition for identification experiments with ICAI, due to safety reasons direct control with ICAC is not allowed.

9 Conclusions and Future Work

This work presents a systematic approach to control system design, which is aimed at industrial processes and use by process engineers. It concentrates on the standardised development of PID based control systems running from linear singlevariable to nonlinear multivariable PID control systems. The applicability of this approach has been verified by a software prototype which was applied to a nonlinear multivariable laboratory process. In the following sections, conclusions are drawn and recommendations for future work are provided.

9.1 Conclusions

This research was motivated by interviews in industry in order to extract the main requirements for the new approach and the practical examination of available software tools. It turned out that the design of nonlinear and multivariable processes in particular are difficult for the average industrial user because currently available control system design software is difficult to use requiring specific knowledge and offering an overwhelming wealth of functions. The efforts to bridge the gap between academic equipment and practical needs by offering special training for non-expert users have not succeeded due the lack of time in industry to get familiar with the software tools provided.

Hence it was found that an easy to follow, intuitive and stepwise control system design procedure for industrial processes is missing, which is aimed at control system design to satisfy industrial needs. This is why this project was formulated and the three main requirements were formulated:

- The user is guided through an intuitive standard control system design path especially in the case of nonlinear or multivariable control system design tasks.
- Only a few control system design methods with easy parameterisation are provided.
- The control system design task is integrated into a block-oriented simulation environment, such that the user can utilise one environment for identification, simulation and control system design.
- The resulting control system has at least an acceptable control behaviour also in case of complex nonlinear multivariable processes to be controlled.

Many steps were necessary towards the realisation of a software prototype for control system design that fits to these requirements. The main novelties that resulted from this contribution are:

An *Industrial CACSD* (*ICACSD*) scheme has been proposed for the solution of practical control system design tasks in the process industry, which consists of a model evolution scheme and a standardised *CACSD* procedure oriented at the industrial users thinking of start simple, add complexity only if necessary. The model evolution scheme reflects the traditional way of doing control system design in a systematic way. The standard *CACSD* procedure supports a constrained control system complexity that can be handled easily and it provides a good reproducibility of the results gained, by restricting the variety of possible solutions. The feasibility of the proposed *ICACSD* approach was tested on simulated processes and a pilot plant.

For the design of nonlinear and multivariable control systems an industrial standardised controller design procedure has been designed. The pragmatic procedure offers a transparent approach to nonlinear and multivariable control system design as long as the process can be described by the introduced models consisting of nonlinear static and linear dynamic blocks.

In order to validate the work the proposed control design procedure has been made accessible in the form of a software prototype with an ergonomically designed graphical user interface allowing easy application of the developed methods. The prototype realisation for the *Industrial Computer Aided Control (ICAC)* toolbox supplies the new structured approach to control system design for industrial processes within a block-oriented simulation environment.

9.2 Future Work

The main issues of the requirements analysis have been addressed within this work, by the provision of a prototypical tool.

Consequently the realised software prototype *ICAC* can be applied directly by even inexperienced users, who look for quick and efficient solutions for control system design. However, the field of control system design for industrial users is still wide and worth exploration. Its potential in industry is still enormous. This final section points the way to possible extensions to this work, and to the *ICACSD* system as a whole and its use in further developments:

- *Improved support of nonlinear multivariable control system design.* The support for the design of nonlinear multivariable control systems has not been implemented completely. It is possible to implement new *ICAC* blocks to support the design of multivariable control systems.

- *Ease of use*. The prototype software could still be improved with respect to the aim of being self-explanatory. For example, the utilisation of tool tips and the display of images on pushbuttons offer possibilities to better assist the user. A data base support system would be beneficial such that it takes care of an automatic comparison between control systems.
- *Completion of the ICACSD system*. This work represents a substantial part of the collaborative research project between the University of Glamorgan and the Fachhochschule Hannover. Overall, the collaborative project aims at making the *ICACSD* approach accessible to engineers, with little or no experience in control engineering in order to replace inefficient control strategies thus leading to a better use of resources.

References

ACKERMANN, J, (1983): *Abtastregelungen*, Springer Verlag, ISBN

ALLGÖWER, F,T A BADGWELL, S J QIN, J B RAWLINGS and S J WRIGHT (1999), *Nonlinear Predictive Control and Moving Horizon Estimation – An Introductory Overview.* eds: Paul M. Frank, Advances in Control – Highlights of ECC'99, Springer-Verlag Berlin, pp. 391-449

ANDERSON, E, Z BAI, C BISCHOF, S L BLACKFORD, J DEMMEL, J DONGARRA, J Du CROZ, A GREENBAUM, S HAMMARLING, A McKENNEY and D SORENSEN, (1999), *LAPACK user's guide* (third edition). Society for industrial and applied mathematics, Philadelphia

ANDERSSON, M, (1989), *An Object-Oriented Language for Model Representation,* IEEE Workshop on CACSD, Danish Automation Society, pp. 8-14

ASTRÖM,K J and T HÄGGLUND, (1984), *Automatic Tuning of Simple Regulators,* 9th Triennial World Congress of IFAC, Budapest, vol.4, pp. 1867-1872

ASTRÖM,K J and T HÄGGLUND, (1988), *Automatic Tuning of PID Controllers,* ISA

ASTRÖM,K J, C C HANG, P PERSSON and W K HO, (1992), *Towards Intelligent PID-Control,* Automatica, vol. 28, no. 1, pp. 1-9

ASTRÖM,K J, *et al.*, (1993), *Automatic Tuning and Adaption for PID Controller: A Survey,* Control Engineering Practise, vol. 1, no. 4, pp. 699-714

ASTRÖM,K J and T HÄGGLUND, (1995), *PID-Controller: Theory, Design and Tuning,* ISA

ASTRÖM,K J and T HÄGGLUND, (1996), *PID Control* in The Control Handbook, pp.189-209

BARKER, H A, (1988), *Graphical Environment for Computer Aided Control System Design,* Preprints of 4th IFAC Symposium on Computer Aided Design in Control Systems, CADCS '88, Beijing, P. R. China, pp. 231-236

BARKER, H A, M CHAN, P W GRANT, L T HARVAY, C P JOBLING and P TOWNSEND, (1990), *The Impact of Recent Developments in User-Interface Specification, Design and Management on the Computer-Aided Design of Control Systems,* 11th World Congress of IFAC, pp. 393-398

BARKER, H A (ed.), (1991), 5th IFAC Symposium on Computer Aided Design in Control, Swansea, UK

BARKER, H A, M CHAN, P W GRANT, C P JOBLING and P TOWNSEND (1993), *Open Architecture for Computer-Aided Control Engineering,* IEEE Control System Magazine, vol.13. no. 2, pp. 17-27

BARKER, H A, P W GRANT, L T HARVAY, C P JOBLING and P TOWNSEND (1996), *An Approach to Project Management in Computer Aided Control Engineering,* Control Engineering Practise, vol.4, no. 4, pp. 441-453

BENVENISTE, A and K J ASTRÖM, (1993), *Meeting the challenge of computer science in industrial applications of control: An introductory discussion to the special issue,* Automatica, vol. 29, no. 5, pp. 1169-1175

BHAT, N, T J McAVOY, (1990), *Use of neural nets for dynamic modelling and control of chemical process systems*, Comput. Chem. Eng., vol. 14, pp. 573.

BIALKOWSKI, W L, (1994), *Dreams Versus Reality: A View from Both Sides of the GAP*, Pulp and Paper, vol. 94:11

BORRIE, J A, (1986), *Modern Control Systems, A Manual of Design Methods*, Prentice-Hall, Inc., ISBN 0-13-590282-2

BROWN, L J, (1993), *Prediction and Adaptation in PID controller design*, 32nd IEEE Conference on Decision and Control, pp. 1575-1580

CAMACHO, E F and C BORDONS, (1999), *Model Predictive Control*, Springer Verlag

CAMERON F and D E SEBORG, (1983), *A self-tuning controller with a PID structure*, Int. J. Control, vol. 30, pp. 401.

CELLIER, F E, (1991), *Continuous System Modeling*, Springer-Verlag, ISBN 0-387-97502-0

CHARWAT, H J, (1996), *Farbkonzept für die Prozeßführung mit Bildschirmen*, atp – Automatisierungs-technische Praxis, vol. 38, no. 5, pp. 50-53

CHEN, G, (1996), *Conventional and fuzzy PID controllers: An overview*, International Journal Intell. Contr. Syst., vol 1, no.2, pp.235-246

CHEN, J, T C HUANG,(2004), *Applying neural networks to on-line updated OID controllers for nonlinear process control*, Journal of Process Control, vol.14, pp. 211-230.

CHIDAMBARAM, M and R PADMA SREE, (2003), *A simple method of tuning PID controller for integrator/dead-time processes*, Computers and Chemical Engineering, vol. 27, pp. 211-215

CHOI, Y S , A D WHITTAKER, D C BULLOCK, (1996), *Predictive neuron-fuzzy controller for multivariable process control*, Trans. Am. Soc. Agric. Eng., vol. 39, no. 4, pp. 1535-1531

CIPRIANO, A and D SAEZ, (1997), Design of fuzzy model based predictive controllers and its application to an inverted pendulum, 6th IEEE International Conference on Fuzzy Systems, vol. 2, pp. 915-919

CLARKE, D W and P J GAWTHROP, (1979), Self-tuning control, in: Proc. IEE, Pt-D, vol. 126, pp. 633.

CLARKE, D W, C MOHTADI and P S TUFFS, (1987), Generalised Predictive Control – *Part I. The Basic Algorithm*, Automatica, vol.23, pp. 137-148

CLARKE, D W and R SCATTOLINI, (1992), *Constrained Receding Horizon Predictive Control*, IEEE Proceedings D, vol. 138, pp. 347-354

COMINOS, P and N MUNRO, (2002), PID controller: recent tuning methods and design to specification, IEE Process Control Theory and Application, vol.149, no.1, pp. 46-53

CUTLER, C R and B L RANAKER, (1980), *Dynamic Matrix Control – A Computer Control Algorithm*, Proceedings of the Joint Automatic Control Conference, San Franciso, USA

CUTLER, C R, A MORSHEDI and J HAYDEL, (1983), *An Industrial Perspective on Advanced Control*, AIChE Annual Meeting, Washington, D C, USA

DEMRCIOGLU, H and P J GAWTHORP, (1991), *Continuous Time Generalised Predictive Control*, Automatica, vol. 27, pp.55-74

DE KEYSER, R M C and A R VAN CUAWENBERGHE, (1985), *Extended Prediction Self-adaptive Control*, Proceddings of IFAC Symposium on Identification and System Parameter Estimation, York, UK, pp. 1255-1260

DE KEYSER, R M C, PH G A VAN DE VELDE and F G A DUMORTIER, (1988), *A Comparative Study of Self-Adaptive Long Range Predictive Control Methods*, Automatica, vol. 24, pp. 149-163

DONGARA, J J, J R BUNCH, C B MOLER and G W STEWART, (1979), *LINPACK User's Guide*, Society for industrial and applied mathematics, Philadelphia

DOUGHERTY, D and D COOPER, (2003), *A practical multiple model adaptive strategy for multivariable model predictive control*, Control Engineering Practise, vol. 11, pp. 649-664

DUTTON, K, S THOMPSON, B BARRACLOUGH, (1997), *The Art of Control Engineering*, Addison Wesley Longman Ltd., ISBN 0-201-17545-2

EATON, J W and J B RAWLINGS, (1992), *Model Predictive Control of Chemical Processes*, Chemical Engineering Science, vol. 47, pp. 705-720

ELMQVIST, H, (1994), *DYMOLA user manual*, DynasimAB

ELMQVIST, H, S E MATTSSOB and M OTTER, (1998) *Modelica- An International Effort to Design an Object-Oriented Modeling Language*, Summer Computer Simulation Conference '98, Reno, Nevada, USA

ENDER, D B, (1993), *Process Control Performance Not as Good as You Think*, control engineering, vol. 40, pp. 180-190

EYKOHOFF, P, (1974), *System Identification*, Wiley, London, ISBN 0-471-24980-7

FOSS, B A, B LOHMANN and W MARQUARDT, (1998), *A Field Study of the Industrial Modeling Process*, Journal of Process Control, vol. 8, no. 5/6, pp. 325-338

FREDERICK, D K, C J HERGET, R KOOL and M RIMVALL, (1992), *ELCS – The Extended List of Control Software*, Dept. of Automatic Control, Swiss Federal Institute of Technology (ETH), Mar., no.5

FROESE, T, (1995), *Übergeordnete Regelung chemischer Verfahren mit Fuzzy Control und Künstlichen Neoronalen Netzen*, in H.-J. Zimmermann (ed.) 'Neuro + Fuzzy', VDI-Verlag, pp. 102-118

FULI, W, L MINGZHONG, Y YINGHUA, (1997), *Neural network pole placement controller for nonlinear systems through linearisation*, American Control Conference, pp. 1984.

FUNK, G P E, (1994), *Applying Online Computer-Aided Control System Design to the Manufacturing Industry Enhances Control Performance*, IEEE/IFAC Symposium on CADCS, pp. 429-434

FRIEDLAND, B, (1986), *Control System Design An Introduction to State Space Methods*, McGraw-Hill Book Company, ISBN 0-07-022441-2

GAGNON, E, A POMERLEAU, A DESBIENS, (1999), *Mu-synthesis of robust decentralised PI controllers*, IEE Proceedings-Control Theory and Applications, vol. 146, pp. 289.

GARBOW, B S, J M BOYLE, J J DONGARRA and C B MOLER, (1977), *Matrix Eigensystem Routines-EISPACK Guide Extensions*, Springer-Verlag, New York

GARCIA, C E, D M PRETT and M MORARI, (1989), *Model Predictive Control: Theory and Practise*, Automatica, vol. 25, pp.149-163

GAWTHROP, P J, (1986), Self-tuning PID controllers: algorithm and implementation, IEEE Trans. Autom. Control, vol. 31, pp. 201.

GAWTHROP, K F and J W PONTON, (1996), *Improved Control Using Dynamic Process Models*, TransIChem, vol. 71, no. A1, pp.63-70

GMA-BERICHT 32, (1997), *Rechnergestützter Entwurf von Regelsystemen*, 2. VDI/VDE-Aussprachetag, Kassel

GMA-BERICHT 36, (2001), *Rechnergestützter Entwurf von Regelsystemen*, 3. VDI/VDE-Aussprachetag, Dresden

GRAMS, T, (1998), *Bedienfehler und ihre Ursachen (Teil 1)*, atp – Automatisierungs-technische Praxis, vol. 40, no. 3, pp. 53-56

GRÜBEL, G and C P Jobling, (1994), *Simulation Tool-Abstractions Based on the Neutral Dynamics-Model Description Format DSblock*, IEEE International Conference of Control, pp. 1-6

GRÜBEL, G, (1995), *Regelungstechnische Entwurfsautomation*, at – Automatisierungstechnik, vol. 43, no. 1, pp. 3-13

HÄGGLUND, T and K J ASTRÖM, (1996), *Automatic Tuning of PID Controllers* in The Control Handbook, pp. 817-826

HAHN, V and G NÖTH, (1993), *Regelungstechnische CAE-Systeme in der Anwendung*, Aussprachetag "Rechnergestützter Entwurf von Regelungssystemen" des VDI/VDE-GMA FA 5.6 "MSR-Rechnerwerkzeuge"

HAHN, V and G NÖTH, (1997), *Anforderungen an regelungstechnisches CAE in der chemischen Verfahrenstechnik*, VDI/VDE-GMA Jahrbuch, VDI-Verlag, pp. 260-280

HANG, C C, T T TAY and V U VASNANI, (1991), *Auto-tuning of multivariable decoupling controllers*, IFAC Intelligent Tuning and Adaptive Control, pp.115-120

HARRIS, T A, *et al.*, (1999), *A Review of Performance Monitoring and Assessment Techniques for Univariate and Multivariate Control Systems*, Journal of Process Control, vol. 9, no. 1, pp. 1-17

HENSEL, H, (1987), *Methoden des rechnergestützten Entwurfs und Echtzeiteinsatzes zeitdiskreter Mehrgrößenregelungen und ihre Realisierung in einem CAD-System*, doctorate thesis TU Darmstadt, Fortschritts-Berichte VDI Reihe 20, Nr.4

HENSEL, H and K H PETER, (1984), *Multivariable PID-Controller Designed by Linear Quadratic State Space Approach*, American Control Conference, pp. 1559-1664

HENSON, M A and D E SEBORG, (1997), *Online Process Control*, Prentice Hall

HO, M T, A DATTA and S P BHATTACHARYYA, (1998), *Design of P, PI and PID Controllers for Interval Plants*, Proceedings of the American Control Conference, Philadelphia, USA, pp. 2496-2501

HOFFMANN, N, (1992), *Simulation neuronaler Netze: Grundlagen, Modelle, Programme in Turbo Pascal,* Vieweg Verlagsgesellschaft, 2. Auflage, ISBN 3-528-15140-4

HOPGIELD, J J, (1982), *Neural Network and Physical Systems with Collective Computational Abilities,* Proceedings of the National Acedemy of Science, USA, vol. 79, no. 4, pp. 2554-2558

HORNIK , K, M STINCHCOMBE, H WHITE, (1989), *Mutliplayer feedforward neural networks are universal approximators,* Neural Networks 2, pp. 359.

HORTSCH, A and K SCHLÜTER, (1996), *Matlab statt C und Fortran – Rechenkünstler,* mc extra, pp. 36-41

ISERMANN, R, (1991), *Digital Control Systems,* Vol. II, Springer-Verlag, ISBN 3-540-16596-7

JOBLING, C P, P W GRANT, H A BARKER and P TOWNSEND, (1994), *Object-Oriented Programming in Control System Design: A Survey,* Automatica, vol. 30, no. 8, pp. 1221-1261

JOBLING, C P, (1996), *Advances in Computer Aided Control Systems Designs,* Computer and Control Division, Colloquium about Advances in CACSD, vol. 61, pp 1/1-1/6

KALMAN, R E and T S A ENGLAR, (1966), *User's Manual for the Automatic Synthesis Program,* NBS #: 70400375

KING, R A and J O GRAY, (1986), *A Methodology for the Design and Implementation of Graphical Man-Machine Interfaces,* CAD, pp. 419-425

KINNEBROCK, W, (1992), *Neuronale Netze: Grundlagen, Anwendungen, Beispiele, Oldenbourg Verlag,* ISBN 3-486-22103-5

KOHONEN, T, (1982), *Self-organized Formation of Topologically Correct Feature Map,* Biological Cybernetics, vol. 43, pp. -69

KÖRNER, S and R SCHUMANN, (1996), *Inbetriebnahmeunterstützung bei PLS durch regelungstechnisches CAE,* VDI/VDE-GMA Kongreß, Meß- und Automatisierungstechnik, Baden-Baden, VDI-Berichte Nr. 1282, pp. 93-103

KÖRNER, S, R SCHUMANN and K J BAKER, (1996), *New Approach to Identification for Block-oriented Simulation Environment,* 8th European Simulation Symposium, Genua, Italy, vol.2, pp. 245-246

KÖRNER, S, R SCHUMANN and K J BAKER, (1997), *ICAI – A MATLAB™ Toolbox for Computer Aided Identification,* 11th IFAC Symposium on System Identification, SYSID, Kitakyushu, Japan, pp. 1611-1616

KÖRNER, S, (1999), *A Structured Approach to Identification Techniques for the Analysis of Industrial Processes,* doctorate thesis, University of Glamorgan – School of Technology, Wales

KOUVARITAKIS, B, J A ROSSITER and A O T CHANG, (1992), *Stable Generalised Predictive Control: An Algorithm with Guaranteed Stability,* IEE Proceedings D, vol. 139, pp. 349-362

KRAMER, K and H UNBEHAUEN, (1991), *Predictive Adaptive Control Comparison of Main Algorithm,* Proceedings of 1st European Control Conference, Grenoble, France, pp. 327-332

KRUSE, H, R MANGOLD, B MECHLER and O PENGER, (1991), *Programmierung Neuronaler Netze: Eine Turbo Pascal Toolbox,* Addison-Wesley, ISBN 3-89319-293-X

LARIMORE, W E and D E SEBORG, (1999), *Automated Multivariable System Identification,* Tutorial Workshop presented at the European Control Conference Karlsruhe

LINKO, S and P LINKO, (1998), *Development in Monitoring and Control of Food Processes,* TchemE, vol. 76, Part C, pp. 127-137

LU, J, G CHEN, H YING, (2001), *Predictive fuzzy PID control: theory, design and simulation,* Information Sciences, vol. 137, pp. 157-187

MacFARLANE, A G J, G Grübel and J Ackermann, (1989), *Future Design Environments for Control Engineering,* Automatica, vol. 25, no. 2, pp. 165-176

MAFFEZZONI, C and P ROCCO, (1995), *Robust Tuning of PID Regulators Based on Step-Response Identification,* European Control Conference, Roma, pp. 2477-2488

MATTSSON, S E, M ANDERSSON and K J ASTRÖM, (1993), *Object-Oriented Modeling and Simulation,* CAD for Control Systems, Linkens, D A (ed.), Marcel Dekker, pp. 31-69

MEIER ZU FARWIG, H and H UNBEHAUEN, (1991), *Knowledge-based System Identification,* IFAC Symposium on Identification and System Parameter Estimation, pp. 21-29

MINSKI, M and S PAPERT, (1969), *Perceptron,* MIT Press, Cambridge

MUNRO, N, (1999), *The Systematic Design of PID Controllers for Continuous and Discrete Systems,* IEE Colloquium on Advances in computer-aided control system design, Birmingham, UK

MUNRO, N and M T SOLYEMEZ, (2000), *Fast Calculation of Stabilizing PID Controllers for Uncertain Systems,* IFAC Conference, ROCOND, Prague,

NAHAS, E P, M A HENSON, D E SEBORG, (1992), *Nonlinear internal model control strategy for neural network models,* Comput. Chem. Eng., vol. 16, pp. 1039.

NARENDRA, K S and K Parthasarathy, (1990), *Identification and control of dynamic systems using neural networks,* IEEE Trans. Neural Networks 1, pp. 4.

NÖTH, G and U KEUCHEL, (1996), *Richtlinienentwurf Inbetriebnahmesysteme für Regelungen,* VDI/VDE-GMA Krongreß, Meß- und Automatisierungstechnik, Baden-Baden, VDI-Bericht Nr. 1282, pp. 103-111

NÖTH, G, (1998), *Randbedingungen für den Einsatz von regelungstechnischen Methoden,* VDI/VDE-GMA Kongreß, Meß- und Automatisierungstechnik, Ludwigsburg, VDI Berichte Nr. 1397, pp. 87-102

ORDYS, A W and D W CLARKE, (1993), *A State-Space Description for GPC Controllers,* Int. Journal System Science, vol. 29, pp. 1727-1744

OTTER, M and P Mostermann, (1998), *The DSblock Model Interface,* version 4 – Incomplete Draft, Http://www.Dynsim.de/Modelica/DSblock/index.html

PANG, G, (1994), *A Knowledge Environment for an Interactive Control System Design Package,* Automatica, vol. 28, no. 3, pp. 473-491

PERCOCO, G and L SARTI, (1996), *On the Usage of Graphics in ICAI Systems,* European Conference on Artificial Intelligence in Education, Lisbon, Portugal, pp. 311-332

PERNE, R, (1995), *Erfahrungen mit einem regelungstechnisch orientierten Simulationswerkzeuges,* Schuler, H, eds.,Prozeßsimulation, VCH Verlag, pp. 291-348

PIWINGER, F, (1975), *Regelungstechnik für Praktiker,* VDI Verlag,

PRETT, D M and C E GARCIA, (1988), *Fundamental Process Control,* Butterworth

PROCEEDINGS 14th World Congress of IFAC, (1999), Beijing, P. R. China

PROCEEDINGS 15th World Congress of IFAC, (2003), Barcelona, Spain

POMERLEAU, D, A POMERLEAU, D HODOUIN, E POULIN, (2003), *A procedure for the design and evaluation of decentralised and model-based predictive multivariable controllers for a pellet cooling process,* Computer and Chemical Engineering, vol. 27, pp. 217-233

PSICHOGIOS; D C and L H Ungar, (1991), *Direct and indirect model based control using artificial neural networks,* Ind. Engin. Chem. Res., vol. 30, pp. 2564.

QIN, J S and T A BADGWELL, (1997), *An Overview of Industrial Model Predictive Control Technology,* 5th International Conference on Chemical Process Control –CPC V, (eds: Kantor, J C, Garcia, C E, Carnahan, B) AIChE Symposium Series no. 316,vol. 93, pp. 232-256

QIN, J S and T A BADGWELL, (2000), *An Overview of Nonlinear Model Predictive Control Applications,* (eds: Allgower, F and A Zheng) Nonlinear Model Predictive Control, pp.369-392

RAKE H and M ENNING, (1996), *INTERKAMA '95: Verfahren der Regelungstechnik,* atp – Automatisierungstechnische Praxis, vol. 38,no. 3, pp. 53-57

RAVEN, O and SZYMKAT, (1997), *Modelling and Calibration with Mechatronic Blockset for SIMULINK,* 11th IFAC Symposium on System Identification, SYSID, Kitakyushu, Japan, vol.3, pp. 1643-1648

REINIG, G, R DITTMAR and O GÖTTMANN, (2000), *INTERKAMA 1999: Produkte im Bereich Advanced Process Control,* , atp – Automatisierungstechnische Praxis, vol. 42, no. 4, pp. 46-57

RICHALET, J, A RAULT, J L TESTUD and J PAPON, (1978), *Model Predictive Heuristic Control: Application to Industrial Processes,* Automatica, vol. 14, pp. 413-428

RICHALET, J, (1993), *Industrial Application of Model Based Predictive Control,* Automatica, vol. 29, no. 5, pp. 1251-1274

RIMVALL, M, (1988), *Interactive Environments for CACSD Software,* 4th IFAC Symposium on Computer Aided Design in Control Systems, CADCS '88, , pp. 17-26

ROSENBLATT, F, (1958), *The Percepton: A Probabilistic Model for Information Storage and Organisation in Brain,* Psychological Review, vol. 65, pp. 386-408

ROSSITER, J A and B KOUVARITAKIS, (1993), *Constrained stable generalised predictive control,* IEE Proceedings D, vol. 140, pp. 243-254

ROSSITER, J A, J R GOSSNER and B KOUVARITAKIS, (1996), *Guaranteeing feasibility in constrained stable generalised predictive control*, IEE Proceedings D, vol. 143, pp. 313-323

ROWLAND, J R, (1986), *Linear Control Systems: Modelling, Analysis, and Design*, John Wiley & Sons, ISBN 0-471-03276-X

RUMELHART, D, G HINTON and R WILLIAMS, (1986), *Learning Internal Representation by Error Propagation*, in: Parallel Distributed Processing, MIT Press, pp. 318-362

SCHMID, C, (1993), *Verfügbare Softwaretools Seminar "Rechnergestützte Analyse und Entwurf von Regelungssystemen"*, VDI Seminarhandbuch, VDI-Bildungswerk, Düsseldorf

SCHMID, C and R SCHUMANN, (1993), *Standardised Data Exchange Between CACSD Systems*, 12th World Congress of IFAC, vol. 2, pp. 765-770

SCHMID, C and R SCHUMANN, (1994), *Standardisierte GMA-Datenschnittstelle für regelungstechnsiche CAE-Werkzeuge*, atp – Automatisierungstechnische Praxis, vol. 36, no. 4, pp. 55-56

SCHÖNEBURG, E, F HEINZMANN and S FEDDERSEN, *Genetische Algorithm und Evolutionsstrategien: Eine Einführung in Theorie und Praxis der simulierten Evolution*, Addison-Wesley, ISBN 3-89319-493-2

SCHULER, H, (1992), *Was behindert den praktischen Einsatz moderner regelungstechnischer Methoden in der Prozessindustrie?*, atp – Automatisierungstechnische Praxis, vol. 34, no. 3, pp 116-123

SCHULER, H and P HOLL, (1998), *Erfolgreiche Anwendung gehobener Prozessführungsstrategien*, atp – Automatisierungstechnische Praxis, vol. 40, no. 2, pp 37-41

SCHULER, H, (2003), *Prozessautomatisierung*, Chemie Ingenieur Technik, vol. 75, no. 10, pp. 1402-1405

SCHUMANN, R, (1982), *Digitale parameradaptive Mehrgrößenregelung*, doctorate thesis TU Darmstadt, PDV-Bericht Nr. 190, Kfk-PDV

SCHUMANN, R, S KÖRNER, K J BAKER and M STRICKRODT, (1996), *Shaping CACSD for Practical Use in Process Industry*, 13th World Congress of IFAC, San Francisco, vol. L, pp. 223-228

SCHUMANN, R, (1998), *CAE von Regelsystemen*, atp – Automatisierungstechnische Praxis, vol. 40, pp. 48-62

SILVA, G J, A DATTA and S P BHATTACHARYYA, (2000), *Stabilization of Time Delay Systems*, Proceedings of the American Control Conference, Chicago, USA, pp. 963-970

SINHA, P K, (1984), *Multivariable Control, An Introduction*, Marcel Dekker, Inc., ISBN 0-8247-1858-5

SKOGESTAD, S, (2003), *Simple analytic rules for model reduction and PID controller tuning*, Journal of Process Control, vol. 13, pp. 291-309

SØRLIE, J A, (1994), *In the Interfacing of Software for the Identification of Grey Box Models*, Automatic Control Department of Signals, Sensors and Systems, Royal Institute of Technology, Stockholm, Sweden

SOLYEMEZ, M T, (1999), *Pole Assignment for Uncertain Systems*, UMIST, Control Systems Centre Series

STRICKRODT, M, (1997), *An Integrated Knowledge Engineering Approach to Process Modelling*, doctorate thesis, University of Glamorgan – Dept. of Mech. and Man. Engineering, Wales

SU, H T and T J McAVOY, (1993), *Integration of multilayer perceptron networks and linear dynamic models: a Hammerstein Modeling Approach*, Ind. Engin. Chem. Res., vol. 32, pp. 1927.

SYSKA, B, S KÖRNER, R SCHUMANN, (1999), *User Friendly CACSD for Complex Industrial Processes*, 14th World Congress of IFAC, Beijing, P. R. China, vol. L, pp 409-414

TAKATSU, H, T ITOH and M ARAKI, (1998), *Future Needs for the Control Theory in Industries – Report and Topics of the Control Technology Survey in Japanese Industry*, Journal of Process Control, vol.8, pp. 369-374

TAYLOR, J H and D K FREDERIC, (1984), *An Expert System Architecture for Computer-Aided Control Engineering*, IEEE, vol. 72, pp. 1795-1805

TAYLOR, J H, M RIMVALL and H A SUTHERLAND, (1991), *Future Developments in Modern Environments for CADCS*, IFAC Computer Aided Design in Control Systems, Swansea, UK, pp. 29-40

TAYLOR, J H, *et al.*, (1994), *Panel Discussion on Open CACSD*, CACSD '94, CACCSD newsletter, Online, Internet, http://faith.swan.ac.uk/chris.html/OpenCACSD/archieve/Cocs/CACSD94-Panel.htm

TAYLOR,J H and CHAN C , (1999), An Intelligent Implementation Aid for Industrial Process Control Systems, American Control Conference, San Diego, USA

THAM, M T, F VAGI,. AJ MORRIS and R K WOOD, (1991), *Multivariable and Multirate Self-Tuning Control: A Distillation Column Case Study*, IEE Proceedings-D. ,vol. 138, no. 1, pp. 9-24

TZAFESTAS, S G, (1994), *Fuzzy Systems and Fuzzy Expert Control: An Overview*, Knowl. Eng. Rev., vol. 9, no. 3, pp. 229-268

The MathWork Inc. (1993), *MATLAB user's guide*

The MathWork Inc. (1996), *Building GUIs with MATLAB*

VAN DEN BOOM, A D, (1988), *CADCS Developments in Europe*, 4th IFAC Symposium on Computer Aided Design in Control Systems, CADCS '88, , pp. 51-59

VARSAMIDES, T, S HOPE and C P JOBLING, (1996), *An Object-Oriented Information Model for Computer-Aided Control Engineering*, Control Engineering Practise, vol.4, no. 7, pp. 929-937

VEGA, P and C PRADA, (1991), *Self- tuning predictive PID controller*, IEEE Proc. D, Control Theory Application, vol. 138, no. 4, pp.303-311

VISIOLI, A, (2001 a), *Tuning of PID controllers with fuzzy logic*, IEE Proc. Control Theory Application, vol. 148,no. 1, pp. 1-8.

VISIOLI, A, (2001 b), *Optimal tuning of PID controllers for integral and unstable processes*, IEE Proc. Control Theory Application, vol. 148, pp. 180.

VISIOLI, A, (2004), *A new design for a PID plus feedforward controller*, Journal of Process Control, vol. 14, pp. 457-463)

WANG, L X, (1994), *Adaptive Fuzzy Control*, Prentice-Hall, Inc.

WITTENMARK, B, (1979), *Self-tuning PID Controllers Based on Pole Placement*, Lund Institute Technical Report TFRT-7179,.

YING, H, W SILER, J J BUCKLEY, (1990), *Fuzzy control theory: A nonlinear case*, Automatica, vol. 26, no. 4, pp. 513-520

Yu, C C, (1999), *Auto-tuning of PID Controllers*, Berlin: Springer.

ZAMARRENO, J M, (2002), *Neural predictive control toolbox for CACSD in MATLAB environment*, Preproceedings 15th World Congress of IFAC, Barcelona, Spain

ZHEN-YU, C (ed.),(1988), 4th IFAC Symposium on Computer Aided Design in Control Systems, Beijing, P. R. China

ZHU, Q M and K WARWICK, (2000), A neuro PID controller for complex dynamic plants, IFAC Workshop on Digital Control – past, present and future of PID control, preprints, Teresa, Spain, pp. 59-64

ZHUANG, M and D P ATHERTON, (1994), *PID Controller Design for a TITO System*, IEE Process Control Theory Application, vol. 141, no. 2, pp. 111-120

ZIEGLER, J G and N B NICHOLS, (1942), *Optimum Settings for Automatic Controllers*, Trans. ASME, vol. 64, pp.759-768

Publications

- KÖRNER, S, B SYSKA, R SCHUMANN, (1997), *Regelungstechnische Entwurfswerkzeuge in der industriellen Praxis*, ACHEMA 97, Kurzfassung der Vortragsgruppen Meß- , Reglungs- und Prozesstechnik, Frankfurt am Main, Germany, DECHEMA e.V.
- SYSKA, B, , R SCHUMANN, (1999), *User Friendly CACSD for Complex Industrial Processes*, 14th World Congress of IFAC, Beijing, P. R. China, vol. L, pp 409-414
- SYSKA, B, *ICAC – Eine MATLAB™-Toolbox für den industrietauglichen Entwurf komplexer Regelungen*,(2001), 3. VDI/VDE-GMA Aussprachetag, Rechnergestützter Entwurf von Regelungssystemen, Dresden, Germany, pp. 28-34
- SYSKA, B, *ICAC - A Matlab Toolbox for Industrial Computer Aided Control*, (2003), 15th World Congress of IFAC, Barcelona, Spain, Preproceedings, no. 588

Regelungstechnische Entwurfswerkzeuge in der industriellen Praxis

Steffen Körner, Birga Syska und Reimar Schumann; REPAM, FH Hannover

EINLEITUNG

Bisher werden regelungstechnische Entwurfswerkzeuge in der industriellen Praxis nur als Insellösungen eingesetzt. Dabei könnte eine integrierte und durchgängige regelungstechnische CAE (Computer Aided Engineering) - Unterstützung insbesondere bei der Planung und Inbetriebnahme von Prozeßleitsystemen (PLS) weitreichende Vorteile bringen. Um dieses Ziel zu erreichen, müssen die regelungstechnischen Entwurfswerkzeuge auf industrielle Nutzer zugeschnitten werden, da die heute erhältlichen regelungstechnischen Softwarepakete in dieser Hinsicht keine befriedigende Unterstützung bieten. Das hier vorgestellte Projekt befaßt sich insbesondere mit verfahrenstechnischen Aufgabenstellungen.

INDUSTRIELLE ANFORDERUNGEN AN REGELUNGSTECHNISCHE ENTWURFSWERKZEUGE

In Bild 1 werden die Phasen zur Planung und zum Betrieb eines PLS gezeigt, ausgenommen die Anlagenmontage. Die Aufgabenstellungen für ein regelungstechnischen CAE-Systemen ergeben sich dabei in vier Phasen. In den ersten beiden Phasen *Angebot* und *Planung* ist der effektive Einsatz von regelungstechnischen Entwurfswerkzeugen direkt von der Verfügbarkeit von Prozeß- und PLS-Modellen abhängig. Während PLS-Funktionsmodelle zumindest prinzipiell aus dem realen PLS direkt ableitbar sind, ist der Zugriff auf Prozeßmodelle ein grundsätzliches Problem, da diese Modelle von den PLS-Herstellern vielfach noch nicht bereitgestellt werden können. Zudem existieren für viele Prozesse keine allgemein verfügbaren, simulierbaren Modelle und selbst bei theoretisch verfügbaren Prozeßmodellen bildet der simulative Aufwand ein erhebliches Einsatzhindernis, das nur durch den systematischen Aufbau von allgemein zugänglichen, *standardisierten Prozeßmodellkatalogen* beseitigt werden kann. In den folgenden Phasen *Inbetriebnahme* und *Betrieb* sollten Inbetriebnahmegeräte sowohl den Entwurf als auch die Optimierung von Regelstrategien unterstützen. Da der Prozeß bereits

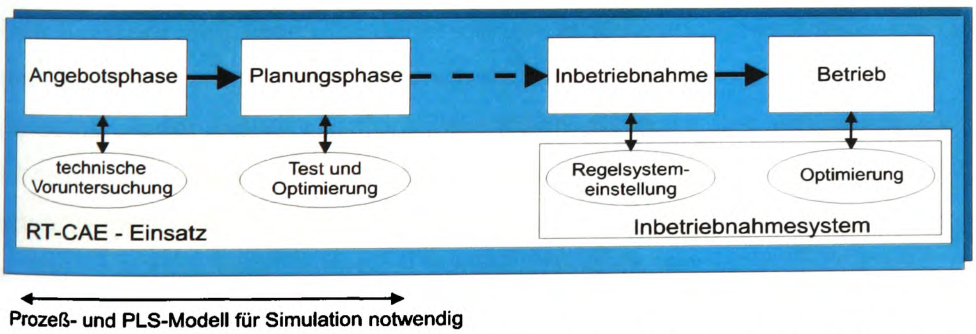


Bild 1. Regelungstechnischer CAE-Einsatz bei Planung und Betrieb von PLS

aufgebaut ist, sollte in diesen Phasen auch die experimentelle Modellbildung intensiv genutzt werden (Nöth und Keuchel 1996).

In der Inbetriebnahmephase einer Anlage kommt der Prozeßleittechnik eine Schlüsselstellung zu, da Komponenten unterschiedlicher Hersteller erstmalig zu einer Funktionseinheit verbunden werden. Allerdings ist es durch die obligatorische Zeitverknappung in der Inbetriebnahmephase meist nicht möglich, den Prozeß optimal einzufahren (Weber 1996). Vielmehr ist die Inbetriebnahme beendet, sobald die im Lastenheft festgesetzten Spezifikationen gerade eingehalten werden. Gerade in dieser Phase können regelungstechnische Entwurfswerkzeuge großen Nutzen bringen. Allerdings sind die in der überwiegenden Mehrzahl an Hochschulen von und für Regelungstechnik-Experten entwickelten regelungstechnischen Softwarepakete für den industriellen Einsatz wenig geeignet, da sie Planern, Inbetriebnehmern und Betriebsleuten eine ungewohnte und nur mit regelungstechnischem Expertenwissen durchführbare Vorgehensweise aufzwingen.

Ein Industrieingenieur benötigt eine *aufgabenorientierte CAE-Unterstützung* für die Lösung schwierigerer Aufgaben wie zum Beispiel der Regelung komplexer Mehrgrößenprozesse durch Bereitstellung verständlicher *Standardlösungswege.* Die mathematischen Modellbeschreibung sollte so gewählt werden, daß auch komplexe nichtlineare Mehrgrößenprozesse nach Möglichkeit als Verschaltung von linearen und einfachen nichtlinearen Blöcken modelliert werden können, die für den Industriepraktiker verständlich bleiben. Insgesamt müssen wenige, leistungsfähige regelungstechnische CAE-Methoden durch leichtverständliche Parametrierung einfach anwendbar und zusätzlich robust sein, sowie einen weiten Einsatzbereich abdecken. Auf Basis derartiger Überlegungen ist ein exemplarisches Konzept zum Einsatz eines regelungstechnischen Entwurfswerkzeuges in der Inbetriebnahmephase verfahrenstechnischer Anlagen entwickelt worden, das die intuitive Vorgehensweise von Inbetriebnehmern systematisiert (Schumann et al. 1996).

ENTWICKLUNG EINES REGELUNGSTECHNISCHES INBETRIEBNAHMESYSTEMS

Das Konzept des regelungstechnischen Inbetriebnahmesystems wurde in der ersten Projektphase mit verschiedenen regelungstechnischen Entwurfswerkzeugen an Laborprozessen mit Erfolg getestet. Aufgrund der unterschiedlichen Stärken der eingesetzten Programme wurde für jede Phase des rechnergestützten Reglerentwurfs ein anderes Programm gewählt, wobei erwartungsgemäß folgende Probleme auftraten:

- verschiedenartige und wenig intuitive Benutzerführung
- komplizierte Parametrierung und unzuverlässige Entwurfsergebnisse
- unterschiedliche Schnittstellen und Dateiformate

Um diese Probleme zu vermeiden, wird derzeit ein MATLAB™-Prototyp eines regelungstechnischen Inbetriebnahmesystems entwickelt (Eröffnungsmenü, siehe Bild 2), der von der Identifikation bis zum Reglerentwurf und -test alle Entwurfsphasen mit aufgabenorientierter, graphischer Benutzerführung unterstützt und in eine blockorientierte Simulationsumgebung integriert ist (Körner et al. 1996). Nach intensiven Tests im Labor und in der Industrie soll dieser Prototyp direkt programmiert werden.

AUSBLICK

Regelungstechnische CAE-Systeme und Inbetriebnahmehilfen besitzen für alle Planungs- und Betriebsphasen eines PLS beträchtliches Rationalisierungspotential. Sie können dabei folgende Verbesserungen bewirken:

- erhöhte Planungssicherheit für regelungstechnische Funktionen
- Verkürzung der Inbetriebnahmezeit für Regelungen

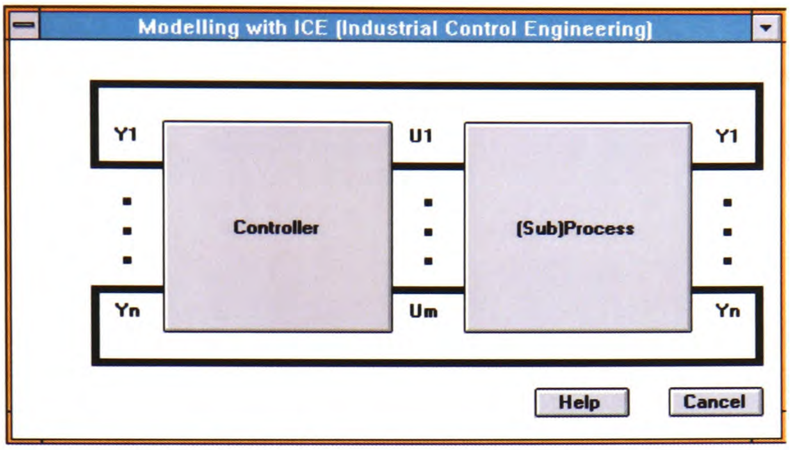


Bild 2. Eröffnungsmenü

Voraussetzung für ihren industriellen Einsatz sind aber:

- effiziente Unterstützung aller regelungstechnischen Entwurfsphasen auch bei komplizierteren Prozessen
- reproduzierbare, verläßliche Entwurfsergebnisse
- Nutzbarkeit für industrielles Inbetriebnahme- und Betriebspersonal

Die Benutzerführung solcher Systeme muß dabei auf die Lösung der industriellen Entwurfsaufgabe zugeschnitten sein. Das hier vorgestellte Konzept für ein industrietaugliches, regelungstechnischen Inbetriebnahmesystem ist ein erster Schritt in Richtung auf eine erweiterte Unterstützung von Planung und Betrieb von PLS durch regelungstechnische CAE-Systeme.

REFERENZEN

KÖRNER, S.; SCHUMANN, R.; BAKER, K. J., 1996, *New approach to identification for blockoriented simulation environments*, 8th European Simulation Symposium, Genua, vol. 2, 245-246.

NÖTH, G.; KEUCHEL, U., 1996, *Richtlinienentwurf Inbetriebnahmesysteme für Regelungen*, VDI/VDE- GMA Kongreß Meß- und Automatisierungstechnik, Baden-Baden, VDI-Bericht Nr. 1282, 103-111.

SCHUMANN, R.; KÖRNER, S.; BAKER, K. J.; STRICKRODT, M., 1996, *Shaping CACSD for practical use in process industry*, IFAC World Congress, San Francisco, vol. L, 228-228.

WEBER, K. H., 1996, *Inbetriebnahme verfahrenstechnischer Anlagen*, VDI-Verlag, Düsseldorf, ISBN 3-18-401367-7.

,505,J,GO,I++,KÖRNER, S; SCHUMANN, R,INBETRIEBNAHMEUNTERSTÜTZUNG BEI PLS DURCH REGELUNGSTECHNISCHES CAE,VDI/VDE- GMA KONGREß 1996 , MEß- UND AUTOMATISIERUNGSTECHNIK,BADEN-BADEN, VDI-BERICHT NR. 1282,1996,SEP.,,,,PP. 93-103,,,

/1/ SCHUMANN, R.: CAE von Regelsystemen. atp - Automatisierungstechnische Praxis, R. Oldenbourg Verlag, 1994, vol. 36, no. 3, pp. 51-59.

14th Triennial World Congress, Beijing, P.R. China

L-5c-01-5

USER FRIENDLY CACSD FOR COMPLEX INDUSTRIAL PROCESSES

Birga Syska, Steffen Körner and Reimar Schumann

Forschungsschwerpunkt REPAM, Fachhochschule Hannover, Ricklinger Stadtweg 120, 30459 Hannover, Germany, e-mail: syska@repam.fh-hannover.de, FAX: +49-511-9296-111

Abstract: In this paper the concept of an industrial CACSD (computer aided control system design) system is outlined which is geared at making available efficient theoretical control system design methods to the process industry. After discussion of general requirements the focus is set on the ICAC (Industrial Computer Aided Control) toolbox developed for MATLAB/SIMULINK™. This toolbox is aimed at industrial users like process engineers or commissioners who are not necessarily specialists in control system design. The ICAC toolbox provides specific methods for the design of linear and nonlinear single and multivariable control systems composed of industrially available function blocks such as PID controllers and typical nonlinear characteristics. Furthermore it allows simplified optimisation of arbitrary industrial control schemes. The ICAC user interface supports 3 user levels with different degrees of functionality adapted to the user's knowledge. *Copyright © 1999 IFAC*

Key Words: CACSD, Industrial Engineering, Process Industry

1. INTRODUCTION

The design of complex control systems for industrial processes is in general still based on experience and trial and error rather than on systematic process analysis and control system design. Typically control systems in the process industry comprise either single PID control loops with clear association of the process input and controlled signal (e.g. heating and temperature) or standard control schemes developed by intuition through the years (e.g. cascade control using an underlying flow control loop). Although many of these control schemes seem to work rather satisfactorily, in most major control systems several poorly tuned or switched off controllers are encountered leading to unsatisfactory process behaviour or manual operation of the process (Hahn and Nöth, 1997). With increasing demands on process efficiency, product quality and environmental compatibility the need for better control and process optimisation leads to the question of how the potential of systematic process analysis and controller design methods developed by control theory over the last decades can be made available to the industrial control engineer to better the situation described. One of the possible answers is the use of industrial computer aided control system design tools tailored to the control design tasks and knowledge level of industrial engineers. The aim is to hide the complexity of theoretical methods under an industrial user interface and to adapt the design results to the realisation means in industrial process control systems.

2. INDUSTRIAL CACSD ENVIRONMENT

Most available CACSD systems have been developed in and for an academic environment. These systems serve mostly as testbeds for newly developed control methods providing a variety of methods and high degrees of freedom for the tuning of the methods. An industrial user, however, needs comprehensive CACSD tools which allow him

- to analyse practical problems encountered in the application of industrial control systems, e.g. by using process identification and simulation tools.
- to design or redesign industrial control schemes with function blocks available in industrial controllers, e.g. with (automatic) control scheme generators.
- to tune industrial control systems according to the requirements of the process operation, e.g. by using numerical or analytical optimisation tools.

For all these tasks the industrial user should not be bothered with the selection of various methods or unnecessary specification of design parameters as required usually in academic CACSD systems. An industrial CACSD system is needed which should provide for every step of the systematic control system design procedure, see Fig. 1, just one default preparametrized design method yielding usable results reliably (rather than optimal results after extensive selection and tuning of the optimal method).

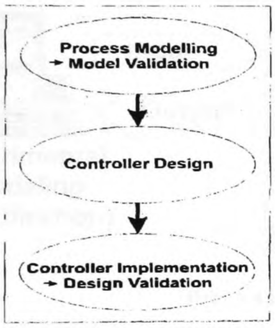


Fig. 1. Systematic Control System Design Procedure

3. THE ICACSD SYSTEM

The above considerations and extensive interviews with industrial users led to the design of the ICACSD (industrial CACSD) system (Schumann, *et al.*, 1996) which comprises in its present state four modules shown in Fig. 2.

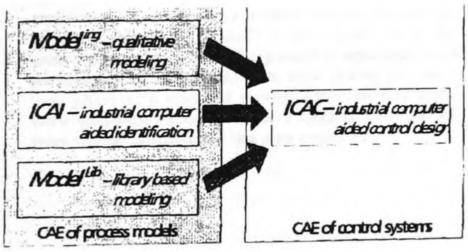


Fig. 2. Industrial CACSD modules

The ICACSD system represents the general frame for an industrial realization of the systematic control system design procedure (Fig. 1). The first user action is the specification of an ICACSD project, for which among other details a user profile and a process characterisation have to be defined, see Fig. 3.

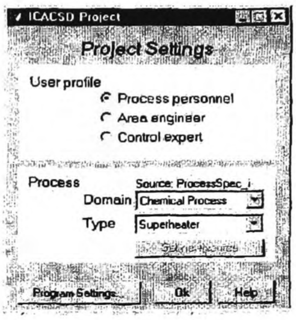


Fig. 3. ICACSD user profile and process specification

The user profile defines the degrees of freedom available for three different user levels: *process personnel* will get a standardized path through the CACSD procedure using preparametrized default methods with almost no degrees of freedom, the *area engineer* can at least access alternative design methods and the *control expert* will have access to all design methods and parameters available. By specifying the process domain and type, the user can add some general information about the process for which the control system is to be designed.

After completion of the project settings, the ICACSD scheme window opens, allowing the access to process modeling for which the user can select four model generation tools, see Fig. 4.

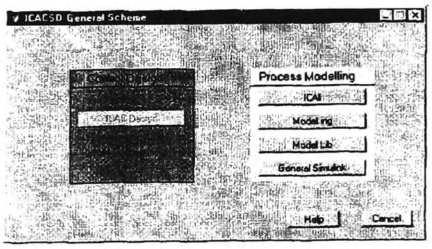


Fig. 4. ICACSD start menu

Besides the general possibility to use SIMULINK™ directly for the specification of a process model, ICACSD provides the choice between three toolboxes for the CAE of process models which are described in the next section.

4. CAE OF PROCESS MODELS

The most important requirement for a systematic analysis and design of control systems is the availability of comprehensive process models. Therefore the CAE of process models is also of primary interest especially for the ICACSD system for which three toolboxes have been specified supporting the generation of process models in different ways and for different phases of process design and operation.

- the *ICAI* (Industrial Computer Aided Identification) toolbox for industrial computer aided identification (Körner and Schumann, 1997). ICAI generates structured linear and nonlinear single- and multivariable process models from measurement data of the process. ICAI is therefore especially useful during start-up and operation of the process where it can serve in addition to analyse the behaviour of the real process

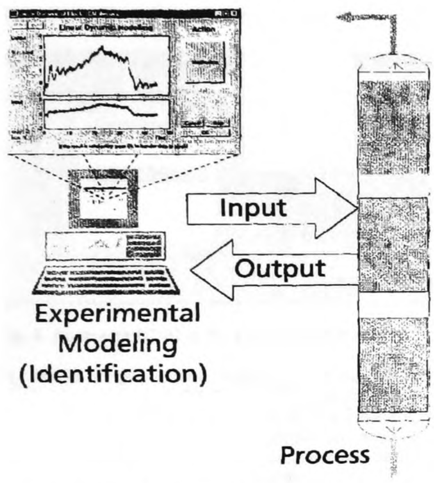


Fig. 5. Process modeling with ICAI

- the *Model^ing* toolbox for the qualitative design of process models using the process knowledge of the industrial user (Strickrodt, *et al.*, 1996). The knowledge engineering approach of Model^ing allows direct interaction with the area expert (i.e. the industrial user) without the need of a knowledge engineer who would normally be responsible for the translation of the unstructured knowledge of the area expert into a formal representation (or process model). As the primary information source of Model^ing is the experience of the area engineer, the Model^ing approach is applicable to a process already in operation or even during the planning phase of a standard process provided that sufficient operation experiences have already been gained from similar processes by the area engineer.

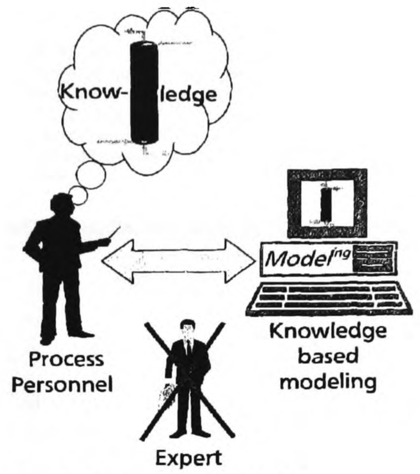


Fig. 6 Knowledge based process Model^ing

- the newly specified *Model^Lib* toolbox aimed at the aggregation of process models from partial (component) models.

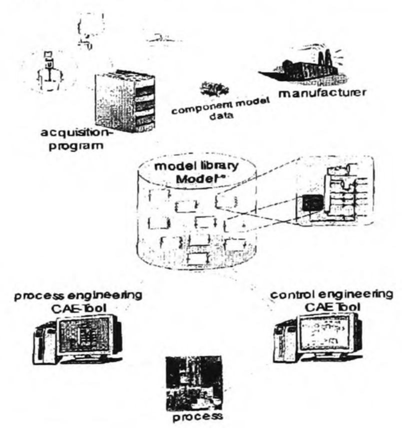


Fig. 7. Library based modeling

The basic idea of Model^Lib relies on the definition of an electronic catalogue of process components like valves, pumps, pipes, superheater etc., including especially dynamic models of the components' behaviour as required for the control system design. Such an electronic catalogue should be supported and filled by the component producers providing all technical information required for planning and maintenance of the process. During the process planning where the process is composed from such components, Model[Lab] should automatically generate dynamic models from the components submodels as required for the control system design for the regarded process area.

5. INDUSTRIAL COMPUTER AIDED CONTROL TOOLBOX – ICAC

After creation of the process model the ICACSD scheme changes its appearances, see Fig. 8. The process modeling menu is replaced by the process model (here: ICAI model), and in addition the control design block becomes accessible guiding the user through the ICAC toolbox. In a predefined sequence the following actions are organised:

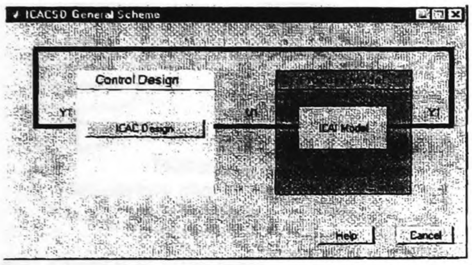


Fig. 8. Accessing ICAC in the ICACSD scheme

1) *Model Preprocessing.* Before designing the control system a general (SIMULINK™) process model may be preprocessed by linearisation or by applying ICAI to create a specific nonlinear single- or multivariable model structure. Original and preprocessed model may be compared by simulation in addition.
2) *Signal association.* In case that a MIMO control system has to be designed the user has to associate every process (model) output to be controlled with the primary process input from which it should be controlled preferably. Thus a symmetric control system structure is organised defining preference pairs of I/O signals.
3) *Control Design.* The control design (ICAC) window is shown in Fig. 9. The design procedure starts with the simplest control system structure (by default), the association of a linear PID controller to every pair of associated process I/O signals. The design method is predefined (numerical or analytical optimisation) and can be changed only by an area engineer or a control expert who, in addition, may also change the normally hidden design parameters. The simulated step responses of the closed loops allow an intuitive evaluation of the control performance which can be modified individually using the sliders for the control action. - If the control performance is not sufficient, alternative control system structures may be tried in a similar way by the user by simply clicking on the checkboxes for nonlinear SISO control and/or MIMO control. In the first case a nonlinear characteristic is added to every PID controller to compensate for process nonlinearities whereas in the second case process couplings are compensated systematically by a decoupling controller network.

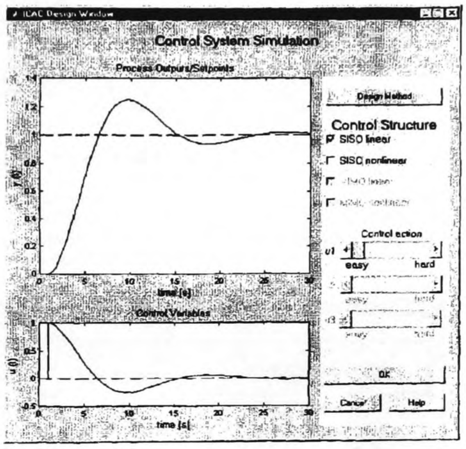


Fig. 9. ICAC design window

The resulting control system, however, consists in every case of linear PID-type control blocks possibly combined with nonlinear characteristic blocks, with which an industrial user is likely to be familiar and which furthermore can be realised with industrial process control systems. Having completed the controller design the user can choose between further evaluation of the design results using the original model (if the design was based on a preprocessed model) or a prototype implementation of the control system using an industrial process control system, or a repetition of the process modeling and/or controller design in case the user is not satisfied with the design results.

6. APPLICATION EXAMPLE

The application of the ICACSD toolboxes ICAI and ICAC will be demonstrated at a laboratory air conditioning plant, Fig. 10, for a which a humidity controller is designed. The plant consist of an air channel where the air flow can be changed by a fan. Air temperature ϑ and relative humidity φ in the mixing chamber at the air outlet are varied by a heater controlled by u_ϑ and by a humidifier controlled by u_φ.

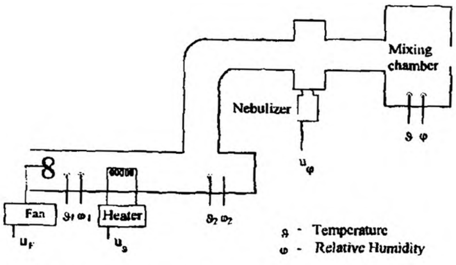


Fig. 10. Laboratory climate plant

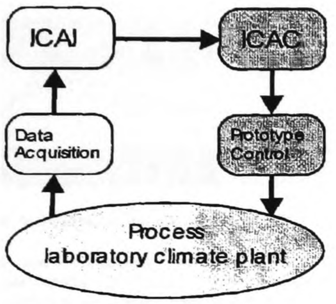


Fig. 11. General ICACSD procedure

The ICACSD procedure as shown in Fig. 11 is carried out in 4 steps

(1) *Data acquisition for identification.*
In the final version of ICACSD data acquisition will be accomplished by means of an interconnected process control system (PCS version) or in the standalone case by making direct use of an A/D-D/A process interface. For the demonstration experiment a separate data acquisition tool was used to collect the experimental data of the process inputs u_ϑ and u_φ and the process outputs ϑ and φ and to stored them in a MATLAB file ready for *ICAI*.

(2) *Process modeling with ICAI.*
Within ICAI the user on level "Process personnel" has just to select the MATLAB file with the measurement data and can then process these data directly and in a simple way to produce a two-input two-output model. The detailed handling within ICAI is described in (Körner and Schumann, 1997) indicating the simplicity of the user interface. Validation of the identified model is done by comparing the measured with the simulated process outputs, see Fig. 12. The submodel connecting u_φ with φ as required for the design of the humidity controller was identified by ICAI as a linear third order process.

$$G(s) = \frac{0.191 - 0.023s + 0.06s^2}{1 + 15.621s + 8.906s^2 + 2.49s^3}$$

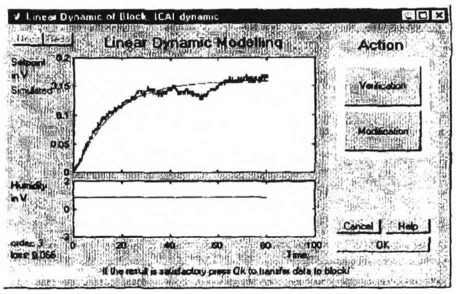


Fig. 12. Model validation with ICAI.

After the completion of the process identification the ICACSD menu changes to the form shown in Fig. 13 where the process model is presented with a canonical two-input two-output model structure.

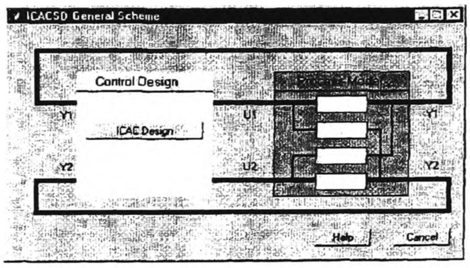


Fig. 13. ICACSD menu with structured process model.

(3) *Controller design with ICAC.*
The model produced by ICAI is selected by default when entering ICAC through the ICACSD menu, Fig. 13. In the simplest case, i.e. on the user level "Process personnel", controller design is simplified to the selection of the inputs and outputs (and thus of the respective submodel) for which the controller is to be designed and the variation of the control action in the ICAC design window, Fig. 9, using the respective sliders until the control performance is acceptable. In the demonstration example just u_φ with φ had to be selected for the design of the humidity controller.

(4) *Prototype control.*
For overall validation of the ICACSD design result the controller is directly applied to the original process. In the final state this will be accomplished either by implementing the controller in the interconnected process control system (PCS version) or in the standalone case using the A/D-D/A process interface. At the moment the prototype controller implementation is still accom-

plished by making use of an external PC based realtime control system. Validation of the design result will be done by checking the simulated control system behaviour against the control behaviour at the real plant as indicated in Fig. 14.

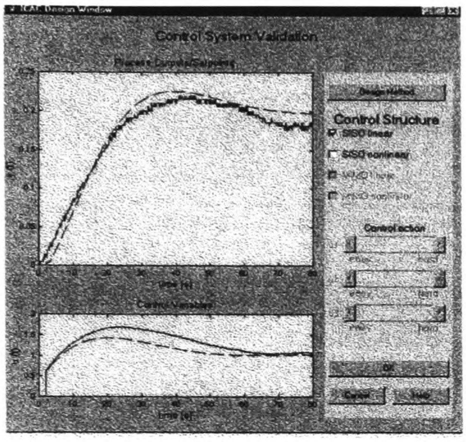


Fig. 14. Validation of the humidity controller

In case the controller performance is not acceptable ICAC and/or ICAI have to be used again to refine the design e.g. by adding nonlinear model and controller parts. As soon as the control performance is satisfactory the ICACSD procedure is finished.

7. CONCLUSION

The ICAC toolbox has been realised in parts as fast prototype. The implementation as SIMULINK/MATLAB™ toolbox is a continuing effort, the first part comprising the above described functionality will be completed in the next year.

The first tests will be run at an absorption process (miniplant) equipped with an industrial process control system serving as interface for data acquisition and as implementation tool for the designed control systems.

Future work will comprise in addition the design of more general control system structures including feedforward control as well as cascade control in arbitrary combinations.

8. ACKNOWLEDGEMENTS

The research was carried out with support of the Volkswagen foundation, REPAM, project 210-70631/9-38-1/92.

9. REFERENCES

Hahn, V., Nöth, G. (1997). Anforderungen an regelungstechnisches CAE in der chemischen Verfahrenstechnik, VDI/VDE-GMA Jahrbuch, VDI-Verlag, Düsseldorf.

Schumann, R., S. Körner, K.J. Baker and M. Strickrodt (1996). Shaping CACSD for practical use in process industry. IFAC World Congress, San Francisco.

Körner, S. and R. Schumann (1997). ICAI – A MATLAB™ toolbox for Computer Aided Identification. IFAC Symposium on System Identification, Fukuoka, Japan.

Strickrodt, M., R. Schumann and K. J. Baker (1996). An integrated knowledge engineering approach to process modeling for CACSD. IFAC World Congress, San Francisco.

ICAC – Eine MATLAB™ - Toolbox für den industrietauglichen Entwurf komplexer Regelungen

Birga Syska

Körting Hannover AG, Badenstedter Straße 56, 30453 Hannover
Tel. +49/(0)511-2921-446
E-Mail: syska@koerting.de

Kurzfassung: Die Industrial Computer Aided Control (ICAC) Toolbox für MATLAB/ SIMULINK™ wurde gezielt für industrielle Anwender wie Prozessingenieure oder Inbetriebnehmer entwickelt, die nicht unbedingt Experten im rechnergestützten Regelsystementwurf sein müssen.
In diesem Beitrag wird die Integration von industrietauglichen Reglerentwurfsverfahren in eine blockorientierte Simulationsumgebung mittels der ICAC Toolbox beschrieben. Für SIMULINK™, die blockorientierte Simulationsumgebung von MATLAB™, sind ICAC CD (Control Design) Funktionsblöcke programmiert worden, die intuitiv wie andere SIMULINK™ Funktionsblöcke gehandhabt werden können und den Benutzer beim Regelsystementwurf durch ein anwenderfreundliches graphisches Benutzerinterface (GUI) unterstützen. Die ICAC Toolbox unterstützt den Entwurf von linearen und nichtlinearen Ein- und Mehrgrößenregelungen und bildet das entwickelte Regelungssystem in industriell vorhandene Funktionsblöcke wie PID-Regler und typische nichtlineare Regelsysteme ab.

Schlüsselworte: Regelungstechnisches CAE, CACSD, Regelsystementwurf

1 EINLEITUNG

In der chemischen Verfahrenstechnik vertraut man immer noch darauf, dass PID-Regler bei gleichermaßen einfacher wie auch transparenter Funktionalität für mindestens 95% aller Anwendungsfälle ausreichend sind. Dies führt diese Praxis jedoch mitunter zu so schlecht eingestellten Reglern, daß das Anlagenpersonal häufig gezwungen ist, die Regelung in Handbetrieb zu nehmen - ein Zustand, der zu Lasten der Produktqualität geht, wertvolle Arbeitskraft kosten und zu unsicheren Betriebszuständen führen kann.

Zudem werden verfahrenstechnische Prozesse immer weiter vermascht, um den Einsatz von Energie und Rohstoffen zu minimieren. Für die Auslegung der dafür notwendigen komplexen Regelsysteme empfiehlt sich der Einsatz regelungstechnischer CAE (Computer Aided Engineering) Werkzeuge.

Grundlage für die Reglersynthese ist dabei ein mathematisches Prozeßmodell, welches das dynamische Verhalten des Prozesses in dem betriebsrelevanten Arbeitsbereich unter Berücksichtigung von Nichtlinearitäten und Kopplungseffekten beschreibt. Während der Reglersynthese werden komplexe Entwurfsmethoden zur Auslegung eines PID-basierten Regelsystems genutzt. In Abhängigkeit vom Benutzerinterface stehen dabei mehr oder weniger Freiheitsgrade zur Verfügung. Die erste Überprüfung des Entwurfsergebnisses erfolgt über die Simulation des Regelsystems mit dem Prozeßmodell, die zweite durch die Erprobung des Reglers am realen Prozeß im Rahmen einer Prototypenregelung. Sobald das entwickelte Regelsystem zufriedenstellend arbeitet, wird der Regler in das industrielle Regelsystem (z.B. ein Prozessleitsystem) implementiert.

2 DIE ICAC TOOLBOX INNERHALB DES ICACSD SCHEMAS

Die ICAC Toolbox ist Teil einer vereinfachten CACSD-Systematik, welche die Synthese komplexer Regelsysteme für industrielle Anwender, die keine regelungstechnischen Experten sind, ermöglicht. Das ICACSD (Industrial Computer Aided Control System Design) Schema wurde entwickelt, um PID basierte Reglersynthese für lineare und nichtlineare SISO (Single Input Single Output) und MIMO (Multiple Input Multiple Output) Prozesse so einfach wie möglich durchführen zu können [1]. Ein Schlüsselelement des ICACSD Schemas ist die Nutzung von Standard-Prozessmodellstrukturen, welche im Modellentwicklungs-Schema, siehe Bild 1, dargestellt sind.

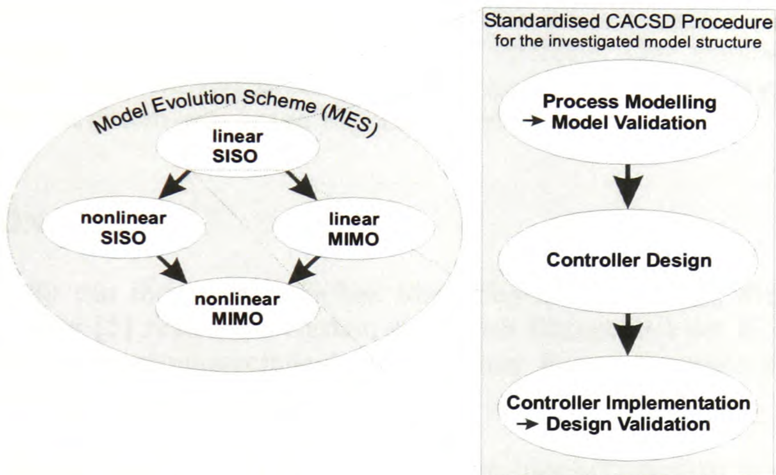


Bild 1. Modellentwicklungsschema und regelungstechnische Entwurfsfolge

Eine Prototypen-Realisierung des ICACSD Schema ist innerhalb der MATLAB/SIMULINK™ Umgebung entwickelt worden und umfaßt, wie in Bild 2 dargestellt, verschiedene Toolboxen für den CAE-unterstützten Prozessmodellentwurf und Regelsystementwurf.

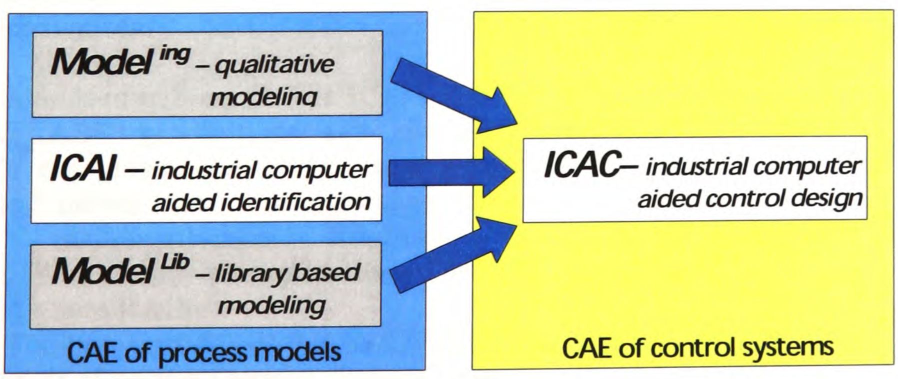


Bild 2. Industrietaugliche CACSD Module

Für den Prozessmodellentwurf stehen drei Toolboxen zur Verfügung, die im Weiteren kurz erläutert werden:

- Die Model^{ing} Toolbox generiert aus dem Prozeßwissen des Anlagenpersonals eine qualitatives Prozessmodell [2].
- Die ICAI Toolbox entwickelt strukturierte lineare und nichtlineare Einzel- und Mehrgrößenmodelle aus Meßwerten des Prozesses [5].
- Die Model^{Lib} Toolbox generiert Prozessmodelle aus einer Bibliothek der Anlagenkomponenten [3].

Innerhalb des ICACSD Schemas werden Prozessmodelle bevorzugt mit Hilfe der ICAI Toolbox generiert, die sie standardmäßig aus linearen dynamischen Blöcken und falls nötig aus nichtlinearen statischen Blöcken zusammensetzt und so z.B. Wiener- oder Hammerstein-Modelle erzeugt. Die ICAC Toolbox entwickelt ein zu den Standard-Prozessmodellen passendes komplementäres Regelsystem innerhalb des ICACSD Schemas, von linearen Eingrößenregelungen bis zu nichtlinearen Mehrgrößenregelungen, abhängig von der Komplexität des Prozessmodells und der geforderten Regelgüte.

Das ICACSD Schema wurde auf die Bedürfnisse von Industrieanwendern zugeschnitten, welche meist über ein geringes regelungstechnisches Wissen verfügen. Die graphische Oberfläche der integrierten Toolboxen zeigt für jeden Schritt der systematischen Reglersynthese einen vorbereiteten Weg, welcher zu einem nutzbaren Ergebnis führt [3].

3 ICAC TOOLBOX

Die Anforderungen für ein industrietaugliches Identifikationswerkzeug, die von [4] formuliert und in der ICAI Toolbox [5] realisiert wurden, sind auch Bestandteil der ICAC Toolbox bei der sinnvollen Unterstützung regelungstechnisch unerfahrener Anwender während des Regelsystementwurfes. Als wesentliche Anforderungen gelten:

- Integration der Regelsystementwurfsaufgabe in eine blockorientierte Simulationsumgebung, so dass der Benutzer nicht zwischen verschiedenen Programmen für Identifikation, Regelsystementwurf und Simulation wechseln muss.
- Bereitstellung weniger, aber robuster Regelsystementwurfsverfahren
- Unterstützung von komplexen Regelsystementwurfsaufgaben bis hin zu nichtlinearen Mehrgrößenregelungen durch standardisierte Vorgehensweisen.

Als Entwicklungsumgebung für die ICAC Toolbox wurde MATLAB™ gewählt, das sich zum Quasistandard für regelungstechnische Software entwickelt hat und neben einer umfangreichen Bibliothek an mathematischen Routinen mit SIMULINK™ auch eine blockorientierte Simulationsumgebung bereitstellt.
Die ICAC Toolbox umfaßt SIMULINK™ Blöcke, sogenannte CD (Control Design) Blöcke, die für die Reglerauslegung von strukturierten Regelsystemen von Ein- oder Mehrgrößenprozessen eingesetzt werden, siehe Bild 3.

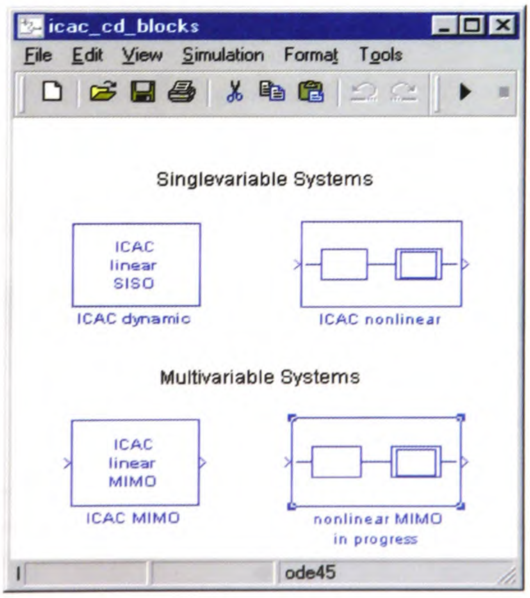


Bild 3. ICAC CD Blockbibliothek

Das anwenderfreundliche graphische Benutzerinterface (GUI) beinhaltet drei Benutzerebenen (Anlagenpersonal – Betriebsingenieur – Regelungsexperte), die mehr oder weniger Freiheitsgrade zur Verfügung stellen. Die Benutzerebene Anlagenpersonal beschränkt die Freiheitsgrade für regelungstechnisch eher unerfahrene Anwender, indem komplizierte Methoden ausgeblendet werden und die Darstellung auf den Zeitbereich beschränkt wird. Auf der Betriebsingenieurebene werden dem Anwender zusätzliche Freiheitsgrade zur Verfügung gestellt (z.B. freie Wahl der ICAC CD Blöcke und selbständige Parametrierung der Auslegungsmethoden). Die Expertenebene ermöglicht den Zugriff auf die komplette in ICAC zur Verfügung gestellten Funktionalitäten (z.B. Benutzung anderer Optimierungs-Toolboxen).

Der Entwurf eines Regelsystemes mit der ICAC Toolbox wird in 3 Schritten durchgeführt, die im Folgenden beschrieben werden:

Schritt 1: Modellvorbereitung
Eine Voraussetzung für den Regelsystementwurf mit Hilfe der ICAC Toolbox ist ein strukturiertes Prozessmodell, wie es die MATLAB™ Toolbox ICAI generiert. Andere SIMULINK™ Prozessmodelle müssen in diese strukturierte Form mit Hilfe der ICAI Toolbox überführt werden.

Schritt 2: Zuordnung der Ein- und Ausgänge
Für den Fall eines Mehrgrößenprozesses erhält der Anwender die Möglichkeit, für jeden Prozessausgang einen bevorzugten Prozesseingang als zugehörige Stellgröße zu bestimmen, mit dessen Hilfe er geregelt werden soll.

Schritt3: Regelsystementwurf mit der ICAC Toolbox
Die in den ICAI Prozessmodellen festgelegte Benutzerebene gibt nun die entsprechende für den Regelsystementwurf angebotene Funktionalität frei. Der Regelsystementwurf wird komplett in dem ICAC CD Block durchgeführt. Für den Entwurf des Regelsystems innerhalb der Anlagenpersonalebene sind nur wenige Standardlösungswege bereitgestellt, die in der Nutzung möglichst unkompliziert sind, so dass der Anwender nur wenige Auslegungsentscheidungen zu treffen hat. Schließlich hat der Anwender am Ende des Entwurfsprozesses durch graphischen Vergleich des simulierten Prozesssignalverlaufs zu entscheiden, ob das Ergebnis ausreicht oder ob weitere Modifikationen nötig sind.

3.1 Regelsystementwurf mit der ICAC Toolbox

Die ICAC CD Blöcke sind eine Erweiterung der SIMULINK™ Blöcke, ihre Handhabung unterscheidet sich nicht von den Standard SIMULINK™ Blöcken. Die noch nicht fertig gestellten CD Blöcke heben sich farblich (blau) von den Standard SIMULINK™ Blöcken ab, auf diese Weise können sie selbst in großen Projekten nicht übersehen werden.

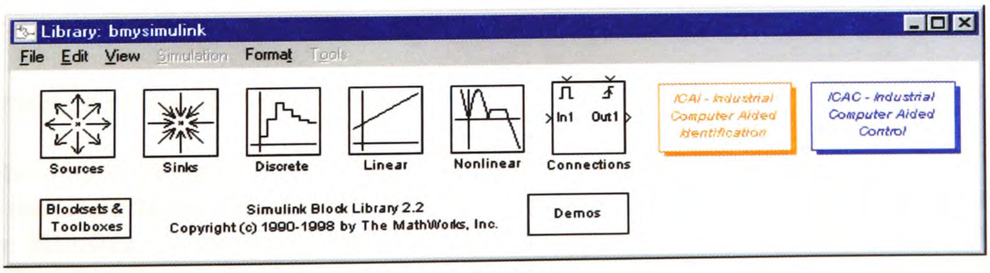


Bild 4. Erweiterte SIMULINK™ Standardbibliothek

Mit einem Doppelklick auf den ICAC Block in der erweiterten SIMULINK™-Standardblockbibliothek (Bild 4) kann das standardisierte graphische Benutzerinterface der

ICAC Toolbox gestartet werden, das den Anwender durch die Regelsystementwurfsaufgabe führt und so gestaltet wurde, dass es ein Minimum an Einarbeitung erfordert.

Der Anwender muss zunächst das Prozessmodell, für das ein Regelsystem entworfen werden soll, in das ICAC Projektfenster ziehen. Die ICAC Toolbox erkennt anhand der mitgelieferten Daten des ICAI Prozessmodells, um welche Art von Prozessmodell – SISO/MIMO, linear/nichtlinear – es sich handelt, das ICAC Benutzerinterface schlägt danach einen ICAC CD Block vor. Dieser ICAC CD Block wird in das ICAC Projektfenster vom Anwender gezogen. Bei Mehrgrössenprozessen muss der Anwender nun die bevorzugte Zuordnung der Stellgrößen zu den Regelgrößen vornehmen. Danach startet der eigentliche Regelsystementwurf. In der Anlagenpersonalebene kann der Anwender während der Entwurfsphase im simuliert geschlossenen Regelkreis nur Einfluss auf die Regeleigenschaften nehmen, indem er den Stelleingriff stärker oder schwächer einstellt (Bild 5). Dann entwickelt die ICAC Toolbox ein Regelsystem, welches vollständig komplementär zum Prozessmodell ist (Bild 6). Innerhalb der Betriebsingenieurebene kann der Anwender ein einfacheres Regelsystem wählen und den Regelsystementwurf starten, indem er nun einen ICAC CD Block mit einer vereinfachten Regelsystemstruktur seiner Wahl in das ICAC Projektfenster zieht, um mit ihm den Regelsystementwurf erneut zu starten. In der Expertenebene kann der Anwender zusätzlich z.B. eine externe Toolbox für die Optimierung des Regelsystems wählen.

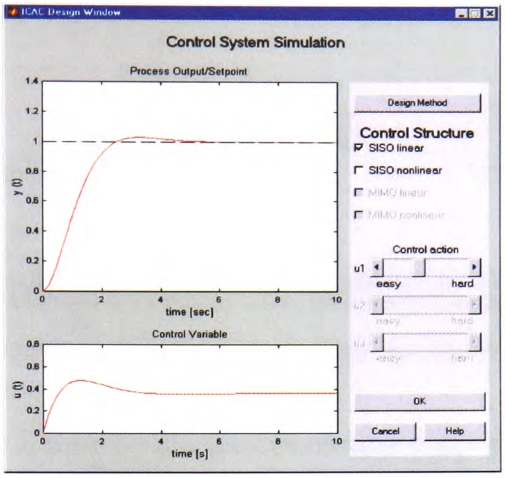


Bild 5. ICAC Design Window

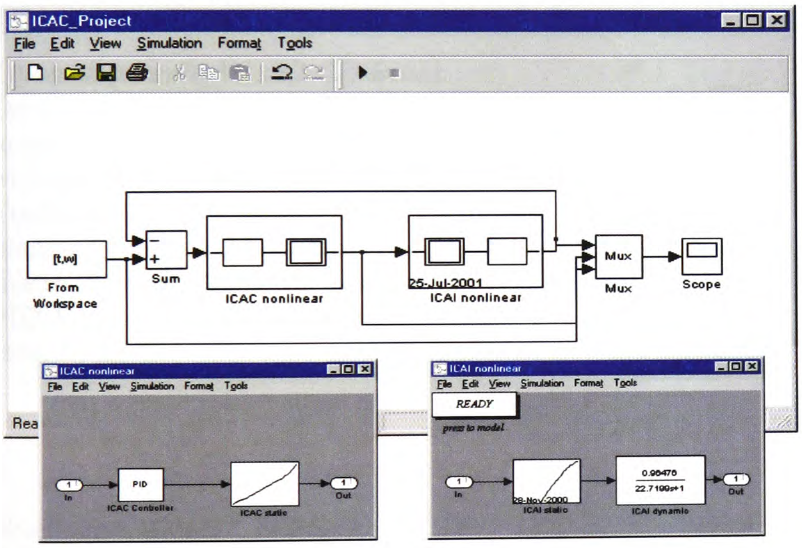


Bild 6. Prozessmodell mit komplementärem Regelsystem

3.2 Regelsystemstrukturen der ICAC Toolbox

Die MATLAB™ Toolbox ICAI generiert vier Arten von Prozessmodellen: linear dynamic SISO, nonlinear SISO, linear dynamic MIMO und nonlinear MIMO. Alle Prozessmodelle bestehen aus nichtlinearen statischen Blöcken (statische Kennlinien) und linear dynamischen Blöcken. Dementsprechend sind die Regelsysteme, die mit Hilfe der MATLAB™ Toolbox ICAC entworfen werden, komplementär zu den ICAI Modellblöcken und können wie folgt beschrieben werden:

Linear Dynamic SISO CD Block
Dieser ICAC CD Block stellt einem linearen Eingrößen-PID-Regler zur Verfügung, der komplementär zu einem linearen ICAI – Eingrößen - Prozessmodell (linear dynamic SISO) ist (Bild 7).

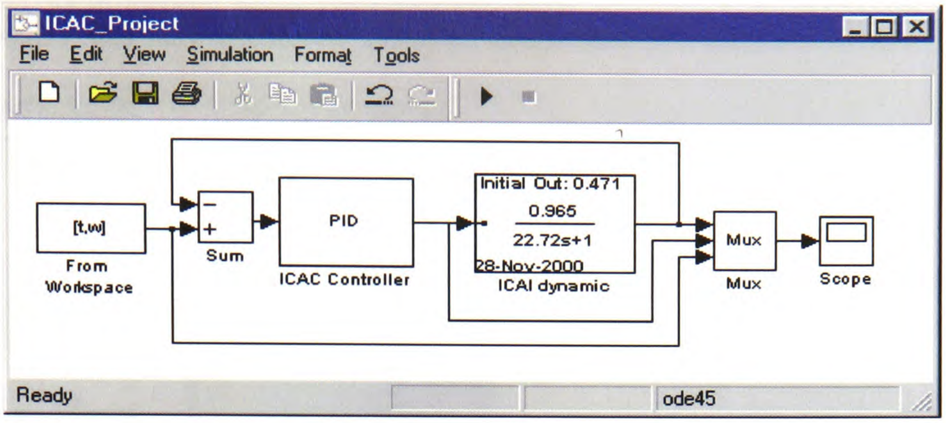


Bild 7. ICAI linear dynamic SISO Prozessmodell mit ICAC linear dynamic SISO Regler

Nonlinear SISO CD Block
Der Nonlinear SISO CD Block beinhaltet eine inverse Kennlinie zur Kompensation eines nichtlinearen statischen Blockes und einen linear dynamischen PID-Regler für den linear dynamischen Teil des Prozessmodells. Das komplementäre ICAI Prozessmodell ist ein nonlinear SISO Modell (siehe Bild 6.).

Linear Dynamic MIMO CD Block
Der Linear Dynamic MIMO CD Block unterstützt den Entwurf eines zum linearen Mehrgrößenprozessmodells der ICAI Toolbox (linear dynamic MIMO) komplementären Regelsystem, das sich aus linear dynamischen PID-Hauptreglern und PID-Entkoppelungsgliedern zusammengesetzt (Bild 8).

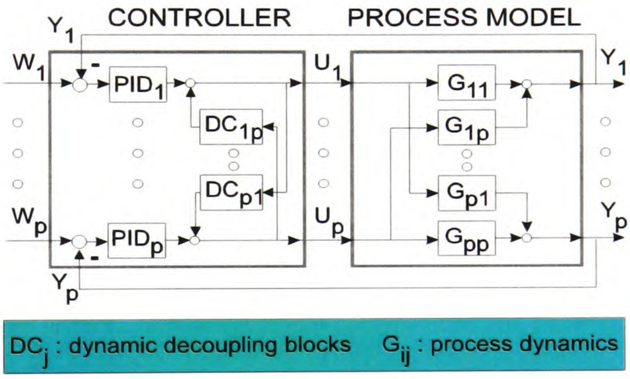


Bild 8. ICAI linear dynamic MIMO Prozessmodell mit ICAC linear dynamic MIMO Regelsystem

Nonlinear MIMO CD Block
Der Nonlinear MIMO CD Block ist eine Kombination des Linear Dynamic MIMO CD Blocks mit Nonlinear SISO CD Blöcken und unterstützt den Entwurf eines zum nonlinear MIMO ICAI Prozessmodell komplementären Regelsystems, welches alle mit ICAC realisierbaren Regelsystemblöcke (lineare PID Regler, lineare Entkopplungsglieder und inverse Kennlinien) beinhaltet (Bild 9.).

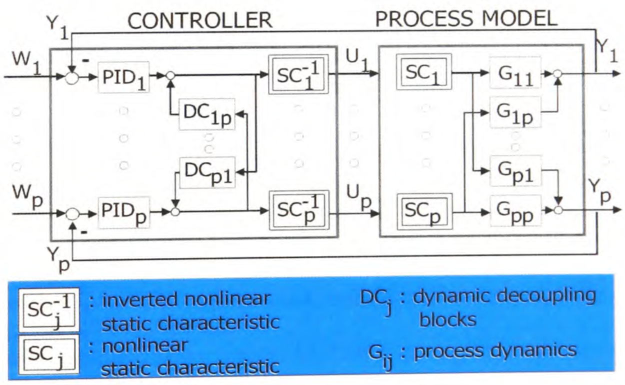


Bild 9. ICAI nonlinear MIMO Prozessmodell mit komplementärem ICAC nonlinear MIMO Regelsystem

Innerhalb der Anlagenpersonalebene steht dem Anwender nur das zum ICAI Prozessmodell genau komplementäre ICAC Regelungssystem zur Verfügung. In den höheren Benutzerebenen (Betriebsingenieur und Regelungsexperte) stehen zur Vereinfachung des Regelsystems auch alle anderen ICAC CD Blöcke mit den dazugehörigen Regelsystemstrukturen dem Anwender zur

Verfügung, so dass z.B. ein nichtlinearer Mehrgrößenprozess auch mit einem Regelsystem geregelt werden kann, welches nur aus mehreren linearen Eingrößen-PID-Reglern besteht (Bild 10).

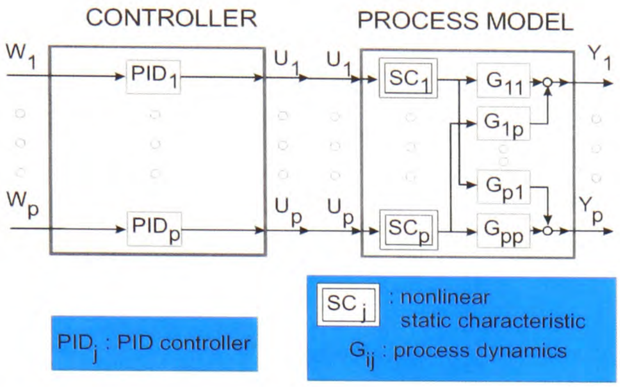


Bild 10. ICAI nonlinear MIMO Prozessmodell mit ICAC linear dynamic SISO Einzelreglern

4 ZUSAMMENFASSUNG

Da die theoretische Regelsystementwicklung sehr aufwendig, mühsam und teuer sein kann, ist die Entwicklung einfach bedienbarer, effizienter Regelsystementwurfswerkzeuge wünschenswert, die eine schnelle und brauchbare Regelsystementwicklung auf Grundlage von Prozessmodellen auch für komplizierte Mehrgrößen-Regelsysteme mit Nichtlinearitäten erlaubt. Für unerfahrene Anwender müssen Standardlösungswege bereitgestellt werden, die so unkompliziert und robust wie möglich sind.

Die beschriebene ICAC Toolbox ist als Bestandteil einer industrietauglichen CACSD Umgebung ausgelegt auf die Bedürfnisse von Industrieingenieuren, um Regelsystementwürfe für nichtlineare Ein- und Mehrgrößenprozesse durchführen zu können. Die entwickelten Regelsysteme sind einfach, unkompliziert und bestehen aus Standardelementen, die in jedem Prozessleitsystem mit Standardfunktionsbausteinen realisiert werden können. Die Übertragung des entworfenen Regelsystems auf das Prozessleitsystem kann daher einfach von Hand durchgeführt werden oder mittels eines OPC-Interfaces, das als Prototyp zur Zeit in Entwicklung ist.

5 LITERATURVERZEICHNIS

[1] SCHUMANN; R., S: KÖRNER, K.J. BAKER AND M. STRICKRODT: Shaping CACSD for Practical Use in Process Industry. 13th World Congress of IFAC, San Francisco, USA, vol. L, pp. 223-228, 1996

[2] STRICKRODT, M.: An Integrated Knowledge Engineering Approach to Process Modelling, Dissertation, University of Glamorgan – Dept. of Mech. and Man. Engineering, Wales, 1997

[3] SYSKA, B, S. KÖRNER,R. SCHUMANN: User Friendly CACSD for Complex Industrial Processes, 14th World Congress of IFAC, Beijing, P. R. China, vol. L, pp. 409-414, 1999

[4] KÖRNER, S. UND R. SCHUMANN: Industrietaugliche Identifikation mit ICAI, GMA-Kongress `98 Mess- und Automatisierungstechnik, VDI Berichte 1397, pp. 119-128,

[5] KÖRNER, S.: A Structured Approach to Identification Techniques for the Analysis of Industrial Processes, Dissertation, University of Glamorgan – School of Design and Advanced Technology, Wales, 1999

15th Triennial World Congress, Barcelona, Spain

ICAC – A MATLAB TOOLBOX FOR INDUSTRIAL COMPUTER AIDED CONTROL

Birga Syska

Körting Hannover AG, Badenstedter Straße 56, 30453 Hannover, Germany
e-mail: syska@koerting.de

Abstract: The Industrial Computer Aided Control (ICAC) toolbox for MATLAB/SIMULINK™ is especially aimed at industrial users like process engineers or commissioners who are not necessarily specialist in control system design. In this paper the integration of industrial control system design functionality into a blockoriented simulation environment by means of the ICAC toolbox is described. For SIMULINK™, i.e. MATLAB™'s simulation environment, ICAC CD (control design) blocks have been programmed, which can be handled intuitively like other SIMULINK™ function blocks supporting a guided tour to control system design based on the simplified use of advanced control design methods and an ergonomically designed graphical user interface (GUI). The ICAC toolbox composes the control system from industrially available function blocks such as PID controllers and typical nonlinear control schemes.

Keywords: Control System Design, User Interface, Industrial Engineering, Process Industry.

1. Introduction

In the process industry 90% of the control systems are realised with conventional PID controllers combined with a variety of nonlinear elements which take care for changes in the process behaviour due to external influences or setpoint changes. Nowadays the design of complex control systems for the process industry (which in many cases can be characterised as multivariable and nonlinear) is still based on experience and trial and error. For a systematic control design several CACSD (Computer Aided Control System Design) tools are available which preferably are developed in or for an academic environment. These tools serve mostly as testbeds for newly developed control methods providing a confusing variety of methods and high degrees of freedom for the use of various methods and the extensive tuning of control systems.

In the process industry controller design tasks have nevertheless to be solved for complex multiple-input multiple-output (MIMO) processes by the process engineers. Using academic CACSD tools these design tasks will lead in general to mathematically complex process models and to the use of powerful theoretical controller design methods which, however, can be understood and handled only by academic control experts - even if the CACSD program is equipped with a sophisticated user guidance system (Meier zu Farwig and Unbehauen, 1991). Moreover, most of the user interfaces of academic CACSD tools were designed to enable extensive tests of various algorithms and methods but do not support efficiently the solution of complex but nevertheless standard industrial controller design tasks.

This paper presents the MATLAB™ toolbox ICAC (Industrial Computer Aided Control) as part of an industrial CACSD (ICACSD) scheme which is tailored to the needs of the engineers in the process industry. First an outline of the ICAC Toolbox as part of the ICACSD scheme is described. Then the functionality of the ICAC Toolbox and the use of the ICAC CD (control design) blocks is outlined.

2. ICAC Toolbox and the ICACSD scheme

The ICAC toolbox is part of a general framework for a simplified CAE approach to the design of complex control systems that can be used by non-expert engineers in industry. The ICACSD (Industrial Computer Aided Control System Design) scheme has been developed to allow the design of PID based control structures for linear and nonlinear SISO (Single Input Single Output) and MIMO (Multiple Input Multiple Output) processes that are as simple as possible and as good as required (Schumann et al., 1996). A key element of the ICACSD scheme is the modelling of process models by a restricted set of standard model structures used in a simple model evolution scheme, see Figure 1.

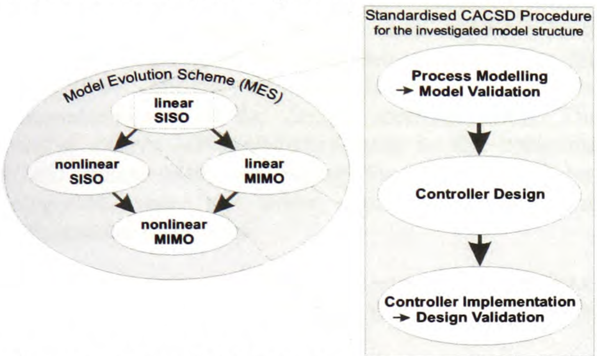


Figure 1. Industrial CACSD model evolution scheme and standardised CACSD procedure

A prototype realisation of the ICACSD scheme was developed within the MATLAB/SIMULINK™ environment and comprises several toolboxes for the generation (CAE) of process models and the design (CAE) of control systems, see Figure 2.

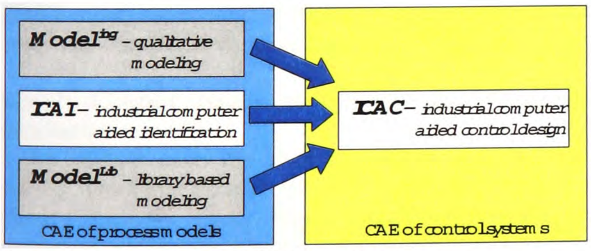


Figure 2. Industrial CACSD modules

For the process modelling three toolboxes are available which are shortly explained in the following:

- The Model^{ing} toolbox has been developed for the qualitative design of process models using the process knowledge of the industrial user (Strickrodt et al, 1996).
- The MATLAB™ toolbox ICAI develops structured linear and nonlinear single- and multivariable process models from measurement data of the process (Körner et al, 1997).
- The planned Model^{Lib} toolbox generates process models from a catalogue of the plant components.

In the ICACSD scheme the process models are preferably produced by the MATLAB toolbox ICAI which generates standardised model structures composed from linear dynamic SISO blocks and, if necessary, nonlinear static SISO blocks possibly resulting in simplified Wiener or Hammerstein type models for SISO and MIMO processes. The ICAC toolbox develops complementary control systems within the ICACSD scheme, from linear SISO to nonlinear MIMO control systems depending on the complexity of the process model and the required control performance.

The ICACSD scheme has been designed to address the industrial user's needs with limited control expertise. So the graphical user interface of all integrated toolboxes provides in the simplest case for every step of the systematic control system design procedure just one default preparametrised design method yielding usable results reliably (Syska et al, 1999).

3. ICAC Toolbox Description

The requirements for an industrial identification tool which were formulated by Körner and Schumann (1998) and realised in the ICAI Toolbox (Körner, 1999), are also requirements of the ICAC Toolbox for the reliable support of inexperienced users during the control system design procedure. The main requirements are:

- Integration of the control system design task into a block oriented simulation environment, such that the user does not have to change between different programs for the individual design tasks of identification, control system design and simulation.
- Only a few, however, robust control system design procedures are available
- Support of complex control system design tasks up to nonlinear MIMO control through standardised procedures.

MATLAB™ was chosen as software environment for the ICAC Toolbox because it became the quasi-standard base for computer aided control system design software. MATLAB™ provides a block oriented simulation environment with SIMULINK™ and also an extensive library of mathematical routines.

The ICAC Toolbox contains a SIMULINK™ blockset comprising CD (control design) blocks for the design of structured SISO and MIMO control systems (Figure 3).

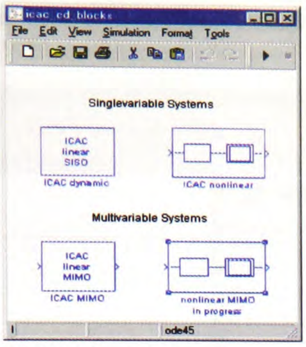


Figure 3. ICAC CD block library

The user-friendly graphical user interface (GUI) includes three user levels (plant personnel - area engineer - control expert) that provides different degrees of freedom. The *plant personnel* user level restricts the degrees of freedom for users with limited control expertise such that complicated methods are hidden, and the presentation of the simulation results is only in time domain. On the *area engineer* level additional degrees of freedom are available for the user (free choice of the ICAC CD blocks, parameterisation of the design methods, etc.). The *control expert* level allows access to the complete ICAC functionality including the unrestricted but integrated use of other control tuning and optimisation toolboxes.

The design of a control system with the ICAC toolbox is carried out in 3 steps that are described in the following:

Step1: Model Pre-processing
The prerequisite for the use of ICAC CD blocks is the availability of structured process models as produced by the MATLAB™ toolbox ICAI. Thus, before designing the control system a general (Simulink™) process model may be pre-processed by applying the MATLAB™ Toolbox ICAI (Körner, 1999) to create a specifically structured linear or nonlinear SISO or MIMO process model. Original and pre-processed model are compared by simulation.

Step 2: Control system structuring by I/O-signal association
In the MIMO case the user has to associate every process (model) output to be controlled with a primary process input from which it should be controlled preferably. Thus a control system structure is organised defining preference pairs of I/O signals.

Step3: Control Design in the ICAC CD blocks
Depending upon the chosen user level ICAC offers different functionality for the control system design. The control design itself is done within the ICAC CD blocks. The design procedure is completely predefined on the *plant personnel* level and makes use of standard design methods such that the user has to make only few basic design decisions. At the end of the design procedure the user decides if the simulated step response of the closed loop simulation is satisfying or more modifications are necessary. The resulting control scheme consists in every case of linear PID control blocks possibly combined with nonlinear static blocks.

3.1 Control System Design with the ICAC Toolbox

The ICAC CD blocks are an extension of the SIMULINK™ blocks, they are handled in the same way as standard SIMULINK™ blocks. The ICAC CD blocks are identifiable from the standard SIMULINK™ blocks by their colour (blue), such that they can be distinguished even in large projects.

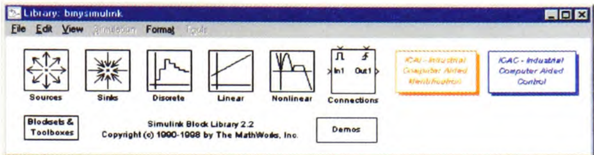


Figure 4. Expanded SIMULINK ™ standard block library

With a double-click on the ICAC block in the extended SIMULINK™ standard block library (Figure 4) the graphical user interface of the ICAC toolbox starts which guides the user through the control system design procedure.

First of all the user has to place the selected ICAC process model for which a control system is to be designed into the ICAC project window. The ICAC toolbox inherits all important data of the ICAI process model, e.g. the type of process model - SISO/MIMO, linear/non-linear -, and proposes a complementary ICAC CD block. This ICAC CD block is placed into the ICAC project window by the user. In case of a multivariable process the user has to define preferable I/O pairs of controlled and manipulated variables. After that the actual control system design starts. On the *plant personnel* level the ICAC toolbox develops a control system, which is completely complementarily to the process model (Figure 5).

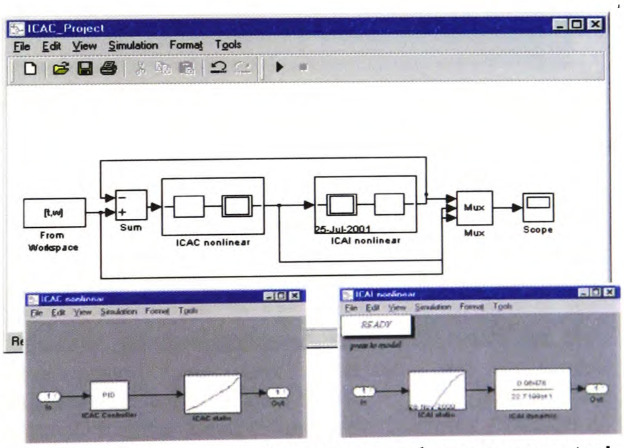


Figure 5. Process model with complementary control system

On this level the user can only modify the control performance using the sliders for stronger or weaker control action in the simulated closed-loop control system (Figure 6).

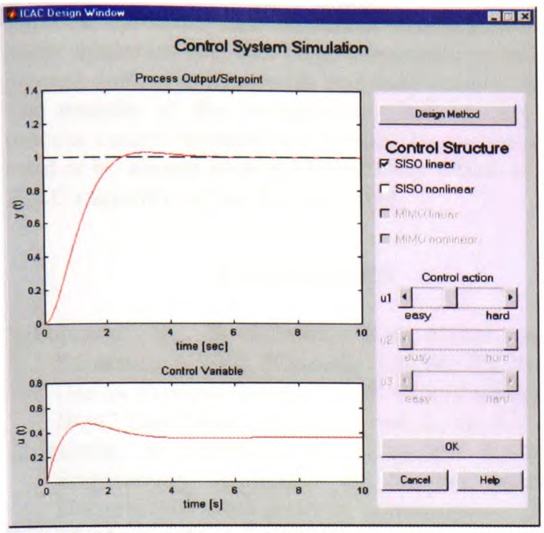


Figure 6. ICAC Design Window

On the *area engineer level* the user can choose a different control system structure and is able to parameterise the optimisation procedure. On the *control expert* level the user can choose in addition external toolboxes for the optimisation of the control system.

3.2 Control System Structures of the ICAC Toolbox

The MATLAB™ ICAI Toolbox generates four kinds of process models: linear dynamic SISO, nonlinear SISO, linear dynamic MIMO and nonlinear MIMO. All process models consist of nonlinear static blocks (static characteristics) and linear dynamic blocks. The control systems, which are designed with the MATLAB™ ICAC Toolbox to use them within conventional process control systems, are correspondingly simple and complementary to the ICAI process models and can be described as follows:

Linear Dynamic SISO CD Block
This ICAC CD block represents a single variable linear PID controller complementary to the ICAI single variable linear process model (linear dynamic SISO) (Figure 7).

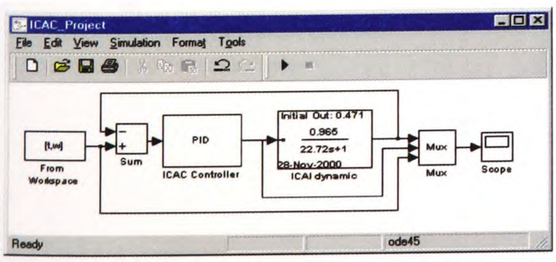


Figure 7. ICAI Linear SISO process model with linear ICAC PID controller

Nonlinear SISO CD block
The Nonlinear SISO CD block includes an inverse characteristic for the compensation of the nonlinear static portion and a linear dynamic PID controller for the control of the linear dynamic part ICAI nonlinear SISO process model (Figure 5).

Linear Dynamic MIMO CD Block
The Linear Dynamic MIMO CD block supports the design of a control system for the linear multivariable process model of the ICAI Toolbox (linear dynamic MIMO), and is composed from linear dynamic PID main controllers and PID decoupling blocks (Figure 8).

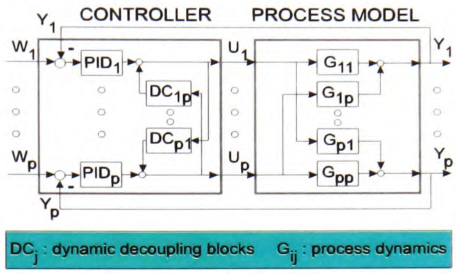


Figure 8. Linear dynamic MIMO process model with linear dynamic control system

Nonlinear MIMO CD block
The Nonlinear MIMO CD block is a combination of the Linear Dynamic MIMO CD block with nonlinear SISO CD blocks and supports the design of a control system complementary to the nonlinear MIMO ICAI process model. It includes all control system blocks realisable with ICAC (linear dynamic PID controllers, decoupling blocks and inverse static characteristic) (Figure 9).

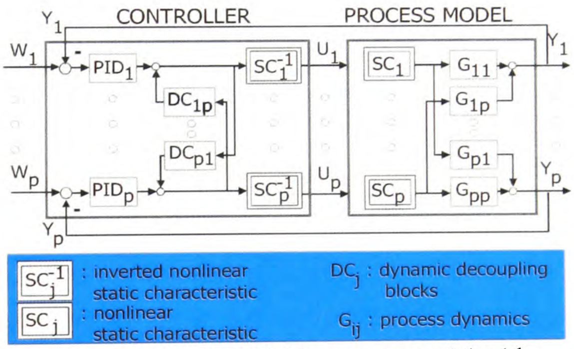


Figure 9. Nonlinear MIMO process model with complementary nonlinear MIMO control system

4. Conclusion

Theoretical control system design can be extensive and expensive and is very often not applicable for industrial control system design. This calls for the development of user friendly and efficient engineering tools which allow a rapid and useful control system design also for complex industrial processes. For inexperienced users standard control system solutions must be provided which are as simple and robust as possible.

The described ICAC Toolbox was developed as a component of an industrial CACSD environment which designs control systems for nonlinear single- and multivariable processes. The developed control systems have a simple structure and consist of standard elements like nonlinear characteristics and linear dynamics and can be implemented in industrial process control system with standard function blocks. The transfer of the designed control system to the process control system can simply be carried out by hand or by means of an OPC interface which is within ICAC currently under development .

REFERENCES

Schumann; R., S. Körner, K.J. Baker and M. Strickrodt (1996). Shaping CACSD for Practical Use in Process Industry. *13th World Congress of IFAC*, San Francisco, USA, **vol. L**, pp. 223-228

Strickrodt, M. (1997). An Integrated Knowledge Engineering Approach to Process Modelling, *Dissertation, University of Glamorgan – Dept. of Mech. and Man. Engineering, Wales*

Syska, B, S. Körner, R. Schumann (1999). User Friendly CACSD for Complex Industrial Processes, *14th World Congress of IFAC*, Beijing, P. R. China, **vol. L**, pp. 409-414

Körner, S. and R. Schumann (1998). Industrietaugliche Identifikation mit ICAI, GMA-Kongress '98 *Mess- und Automatisierungstechnik, VDI Berichte 1397*, pp. 119-128

Körner, S. (1999). A Structured Approach to Identification Techniques for the Analysis of Industrial Processes, *Dissertation, University of Glamorgan – School of Design and Advanced Technology*, Wales